show gsr

To display hardware information on the Cisco 12000 series Gigabit Switch Routers (GSRs), use the **show gsr** command in EXEC mode.

show gsr [**chassis-info** [**details**]]

Syntax Description

chassis-info	(Optional) Displays backplane NVRAM information.
details	(Optional) In addition to the information displayed, this option includes hexadecimal output of the backplane NVRAM information.

Command Modes

EXEC

Command History

Release	Modification
11.2GS	This command was introduced to support the Cisco 12000 series GSRs.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

Use this command to determine the type of hardware installed in your Cisco 12000 series GSR router.

Examples

The following is sample output from the **show gsr** command for a Cisco 12012 router. This command shows the type and state of the card installed in the slot.

```
Router# show gsr

Slot 0  type  = Route Processor
        state = IOS Running  MASTER
Slot 7  type  = 1 Port Packet Over SONET OC-12c/STM-4c
        state = Card Powered
Slot 16 type  = Clock Scheduler Card
        state = Card Powered  PRIMARY CLOCK
```

The following is sample output from the **show gsr chassis-info** command for a Cisco 12012 router:

```
Router# show gsr chassis-info

Backplane NVRAM [version 0x20] Contents -
  Chassis: type 12012 Fab Ver: 1
    Chassis S/N: ZQ24CS3WT86MGVHL
  PCA: 800-3015-1  rev: A0  dev: 257  HW ver: 1.0
    Backplane S/N: A109EXPR75FUNYJK
  MAC Addr: base 0000.EAB2.34FF  block size: 1024
  RMA Number: 0x5F-0x2D-0x44  code: 0x01  hist: 0x1A
```

show gt64010 (7200)

To display all GT64010 internal registers and interrupt status on the Cisco 7200 series routers, use the **show gt64010** command in EXEC mode.

show gt64010

Syntax Description

This command has no arguments or keywords.

Command Modes

EXEC

Command History

Release	Modification
11.2	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

This command displays information about the CPU interface, DRAM/device address space, device parameters, direct memory access (DMA) channels, timers and counters, and protocol control information (PCI) internal registers. The information is generally useful for diagnostic tasks performed by technical support only.

Examples

The following is a partial sample output for the **show gt64010** command:

```
Router# show gt64010

GT64010 Channel 0 DMA:
 dma_list=0x6088C3EC, dma_ring=0x4B018480, dma_entries=256
 dma_free=0x6088CECC, dma_reqt=0x6088CECC, dma_done=0x6088CECC
 thread=0x6088CEAC, thread_end=0x6088CEAC
 backup_thread=0x0, backup_thread_end=0x0
 dma_working=0, dma_complete=6231, post_coalesce_frames=6231
 exhausted_dma_entries=0, post_coalesce_callback=6231

GT64010 Register Dump: Registers at 0xB4000000

CPU Interface:
 cpu_interface_conf   : 0x80030000 (b/s 0x00000380)
 addr_decode_err      : 0xFFFFFFFF (b/s 0xFFFFFFFF)
Processor Address Space :
 ras10_low            : 0x00000000 (b/s 0x00000000)
 ras10_high           : 0x07000000 (b/s 0x00000007)
 ras32_low            : 0x08000000 (b/s 0x00000008)
 ras32_high           : 0x0F000000 (b/s 0x0000000F)
 cs20_low             : 0xD0000000 (b/s 0x000000D0)
 cs20_high            : 0x74000000 (b/s 0x00000074)
 cs3_boot_low         : 0xF8000000 (b/s 0x000000F8)
 cs3_boot_high        : 0x7E000000 (b/s 0x0000007E)
 pci_io_low           : 0x00080000 (b/s 0x00000800)
 pci_io_high          : 0x00000000 (b/s 0x00000000)
 pci_mem_low          : 0x00020000 (b/s 0x00000200)
 pci_mem_high         : 0x7F000000 (b/s 0x0000007F)
```

```
 internal_spc_decode  : 0xA0000000 (b/s 0x000000A0)

 bus_err_low          : 0x00000000 (b/s 0x00000000)
 bus_err_high         : 0x00000000 (b/s 0x00000000)
.
.
.
```

show history

To list the commands you have entered in the current EXEC session, use the **show history** command in EXEC mode.

show history

Syntax Description

This command has no arguments or keywords.

Command Modes

EXEC

Command History

Release	Modification
10.0	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

The command history feature provides a record of EXEC commands you have entered. The number of commands that the history buffer will record is determined by the **history size** line configuration command or the **terminal history size** EXEC command.

Table 86 lists the keys and functions you can use to recall commands from the command history buffer.

Table 86 History Keys

Key	Function
Ctrl-P or Up Arrow[1]	Recalls commands in the history buffer in a backward sequence, beginning with the most recent command. Repeat the key sequence to recall successively older commands.
Ctrl-N or Down Arrow[1]	Returns to more recent commands in the history buffer after recalling commands with Ctrl-P or the Up Arrow. Repeat the key sequence to recall successively more recent commands.

1. The arrow keys function only with ANSI-compatible terminals.

Examples

The following is sample output from the **show history** command, which lists the commands the user has entered in EXEC mode for this session:

```
Router# show history
  help
  where
  show hosts
  show history
Router#
```

Related Commands

Command	Description
history size	Enables the command history function, or changes the command history buffer size for a particular line.
terminal history size	Enables the command history feature for the current terminal session, or changes the size of the command history buffer for the current terminal session.

show idb

To display information about the status of interface descriptor blocks (IDBs), use the **show idb** command in privileged EXEC mode.

show idb

Syntax Description

This command has nor arguments or keywords.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1	This command was introduced.
12.2(15)T	The output of this command was changed to show additional information.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Examples

The following is sample output from the **show idb** command:

```
Router# show idb

Maximum number of Software IDBs 8192.  In use 17.

                        HWIDBs     SWIDBs
Active                       5         14
Inactive                    10          3
Total IDBs                  15         17
Size each (bytes)         5784       2576
Total bytes              86760      43792

HWIDB#1   1   2   GigabitEthernet0/0 0 5, HW IFINDEX, Ether)
HWIDB#2   2   3   GigabitEthernet9/0 0 5, HW IFINDEX, Ether)
HWIDB#3   3   4   GigabitEthernet9/1 6 5, HW IFINDEX, Ether)
HWIDB#4   4   5   GigabitEthernet9/2 6 5, HW IFINDEX, Ether)
HWIDB#5   13  1   Ethernet0 4 5, HW IFINDEX, Ether)
```

Table 87 describes the significant fields shown in the display.

Table 87 show idb Field Descriptions

Field	Description
In use	Total number of software IDBs (SWIDBs) that have been allocated. This number never decreases. SWIDBs are never deallocated.
Active	Total number of hardware IDBs (HWIDBs) and SWIDBs that are allocated and in use.
Inactive	Total number of HWIDBs and SWIDBs that are allocated but not in use.
Total	Total number of HWIDBs and SWIDBs that are allocated.

show idprom

To display the identification programmable read-only memory (IDPROM) information for field-replaceable units (FRUs), use the **show idprom** command in privileged EXEC mode.

show idprom {**all** | *frutype*} [**detail**]

Syntax Description

all	Displays the information for all FRU types.
frutype	Type of FRU for information to be displayed; see the "Usage Guidelines" section for valid values.
detail	(Optional) Displays the detailed display of IDPROM data (verbose).

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(14)SX	This command was introduced on the Supervisor Engine 720.
12.2(17d)SXB	Support for this command on the Supervisor Engine 2 was integrated into Release 12.2(17d)SXB.
12.2(18)SXE	The **module** keyword was modified to support slot/subslot addressing for shared port adapters (SPAs) and SPA interface processors (SIPs), and the optional **clei** keyword was added. The **interface** keyword was replaced by the **transceiver** keyword.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

Valid entries for *frutype* are as follows:

- **backplane**
- **clock** *number*—1 and 2.
- **earl** *slot*—See the following paragraph for valid slot values.
- **module** *slot/port* | {*slot* | *slot/subslot* [**clei**] }—See the following paragraphs for valid values and descriptions.
- **rp** *slot*—See the following paragraph for valid slot values.
- **power-supply**—1 and 2.
- **supervisor** *slot*—See the following paragraph for valid slot values.
- **transceiver** {*slot/subslot/port* | *slot/subslot* [**GigabitEthernet** | **GigabitEthernetWAN**]}
- **vtt** *number*—1 to 3.

The **module** *slot/port* argument designates the module slot location and port number.

Valid values for *slot* depend on the specified interface type and the chassis and module that are used. For example, if you specify a Gigabit Ethernet interface and have a 48-port 10/100BASE-T Ethernet module that is installed in a 13-slot chassis, valid values for the module number are from 1 to 13 and valid values for the port number are from 1 to 48.

The **module** {*slot* | *slot*/*subslot* [**clei**]} syntax designates either the *slot* location alone of the SIP in the chassis (to show information for the SIP only), or the *slot* location of the SIP and the *subslot* location of a SPA installed within the SIP (to display information for a SPA only). Valid values for *slot* depend on the chassis model (2–13), and valid values for *subslot* depend on the SIP type (such as 0–3 for a Cisco 7600 SIP-200 and Cisco 7600 SIP-400). The optional **clei** keyword specifies display of the Common Language Equipment Identification (CLEI) information for the specified SIP or SPA.

Use the **show idprom backplane** command to display the chassis serial number.

Use the **transceiver** *slot*/*subslot*/*port* form of the command to display information for transceivers installed in a SPA, where *slot* designates the location of the SIP, *subslot* designates the location of the SPA, and *port* designates the interface number.

The **interface** *interface slot* keyword and arguments supported on GBIC security-enabled interfaces have been replaced by the **transceiver** keyword option.

To specify LAN Gigabit Ethernet interfaces, use the **show idprom transceiver** *slot*/*subslot* **GigabitEthernet** form of the command.

- To specify WAN Gigabit Ethernet interfaces, use the **show idprom transceiver** *slot*/*subslot* **GigabitEthernetWAN** form of the command.

Examples

This example shows how to display IDPROM information for clock 1:

```
Router# show idprom clock 1

IDPROM for clock #1
  (FRU is 'Clock FRU')
  OEM String = 'Cisco Systems'
  Product Number = 'WS-C6000-CL'
  Serial Number = 'SMT03073115'
  Manufacturing Assembly Number = '73-3047-04'
  Manufacturing Assembly Revision = 'A0'
  Hardware Revision = 1.0
  Current supplied (+) or consumed (-) = 0.000A
```

Table 88 describes the significant fields shown in the display.

Table 88 show idprom Field Descriptions

Field	Description
FRU is	Indicates the type of the field-replacement unit (FRU) to which the information that follows applies.
OEM String	Names the original equipment manufacturer (OEM).
Product Number	A number that identifies a product line.
Serial Number	A number that uniquely identifies the product itself.
Manufacturing Assembly Number	A number that identifies the hardware identification number.
Manufacturing Assembly Revision	A number that identifies the manufacturing assembly number.
Hardware Revision	A number that represents the hardware upgrade.
Current supplied (+) or consumed (-)	Indicated the amount of electrical current that the device supples or uses.

This example shows how to display IDPROM information for power supply 1:

```
Router# show idprom power-supply 1

IDPROM for power-supply #1
  (FRU is '110/220v AC power supply, 1360 watt')
  OEM String = 'Cisco Systems, Inc.'
  Product Number = 'WS-CAC-1300W'
  Serial Number = 'ACP03020001'
  Manufacturing Assembly Number = '34-0918-01'
  Manufacturing Assembly Revision = 'A0'
  Hardware Revision = 1.0
  Current supplied (+) or consumed (-) = 27.460A
```

This example shows how to display detailed IDPROM information for power supply 1:

```
Router# show idprom power-supply 1 detail

IDPROM for power-supply #1
IDPROM image:

  (FRU is '110/220v AC power supply, 1360 watt')

IDPROM image block #0:
  hexadecimal contents of block:
  00: AB AB 01 90 11 BE 01 00 00 02 AB 01 00 01 43 69    ..............Ci
  10: 73 63 6F 20 53 79 73 74 65 6D 73 2C 20 49 6E 63    sco Systems, Inc
  20: 2E 00 57 53 2D 43 41 43 2D 31 33 30 30 57 00 00    ..WS-CAC-1300W..
  30: 00 00 00 00 00 00 41 43 50 30 33 30 32 30 30 30    ......ACP0302000
  40: 31 00 00 00 00 00 00 00 00 00 33 34 2D 30 39 31    1.........34-091
  50: 38 2D 30 31 00 00 00 00 00 00 41 30 00 00 00 00    8-01......A0....
  60: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
  70: 00 00 00 01 00 00 00 00 00 00 00 09 00 0C 00 03    ................
  80: 00 01 00 06 00 01 00 00 00 00 0A BA 00 00 00 00    ................

  block-signature = 0xABAB, block-version = 1,
  block-length = 144, block-checksum = 4542

  *** common-block ***
  IDPROM capacity (bytes) = 256  IDPROM block-count = 2
  FRU type = (0xAB01,1)
  OEM String = 'Cisco Systems, Inc.'
  Product Number = 'WS-CAC-1300W'
  Serial Number = 'ACP03020001'
  Manufacturing Assembly Number = '34-0918-01'
  Manufacturing Assembly Revision = 'A0'
  Hardware Revision = 1.0
  Manufacturing bits = 0x0  Engineering bits = 0x0
  SNMP OID = 9.12.3.1.6.1.0
  Power Consumption = 2746 centiamperes    RMA failure code = 0-0-0-0
  *** end of common block ***

IDPROM image block #1:
  hexadecimal contents of block:
  00: AB 01 01 14 02 5F 00 00 00 00 00 00 00 0A BA    ....._..........
  10: 0A BA 00 16                                     ....

  block-signature = 0xAB01, block-version = 1,
  block-length = 20, block-checksum = 607

  *** power supply block ***
  feature-bits:  00000000 00000000
  rated current at 110v:  2746    rated current at 220v:  2746   (centiamperes)
```

```
  CISCO-STACK-MIB SNMP OID = 22  *** end of power supply block ***

End of IDPROM image
```

This example shows how to display IDPROM information for the backplane:

```
Router# show idprom backplane

IDPROM for backplane #0
  (FRU is 'Catalyst 6000 9-slot backplane')
  OEM String = 'Cisco Systems'
  Product Number = 'WS-C6009'
  Serial Number = 'SCA030900JA'
  Manufacturing Assembly Number = '73-3046-04'
  Manufacturing Assembly Revision = 'A0'
  Hardware Revision = 1.0
  Current supplied (+) or consumed (-) = 0.000A
```

The following example shows sample output for a Cisco 7600 SIP-400 installed in slot 3 of the router:

```
Router# show idprom module 3

IDPROM for module #3
  (FRU is '4-subslot SPA Interface Processor-400')
  OEM String = 'Cisco Systems'
  Product Number = '7600-SIP-400'
  Serial Number = 'JAB0851042X'
  Manufacturing Assembly Number = '73-8404-10'
  Manufacturing Assembly Revision = '09'
  Hardware Revision = 0.95
  Current supplied (+) or consumed (-) = -6.31A
```

The following example shows sample output for the **clei** form of the command on a Cisco 7600 SIP-200 installed in slot 2 of the router:

```
Router# show idprom module 2 clei

FRU             PID                 VID SN          CLEI
--------------- ------------------- --- ----------- ----------
module #2       7600-SIP-200        V01
```

The following example shows sample output for the **detail** form of the command on a Cisco 7600 SIP-400 installed in slot 3 of the router:

```
Router# show idprom module 3 detail

IDPROM for module #3
IDPROM image:

  (FRU is '4-subslot SPA Interface Processor-400')

IDPROM image block #0:

  block-signature = 0xABAB, block-version = 3,
  block-length = 160, block-checksum = 4600

  *** common-block ***
  IDPROM capacity (bytes) = 512  IDPROM block-count = 2
  FRU type = (0x6003,1103)
  OEM String = 'Cisco Systems'
  Product Number = '7600-SIP-400'
  Serial Number = 'JAB0851042X'
  Manufacturing Assembly Number = '73-8404-10'
  Manufacturing Assembly Revision = '09'
```

```
  Manufacturing Assembly Deviation = '00'
  Hardware Revision = 0.95
  Manufacturing bits = 0x0  Engineering bits = 0x0
  SNMP OID = 9.5.1.3.1.1.2.1103
  Power Consumption = -631 centiamperes    RMA failure code = 0-0-0-0
  CLEI =
  VID =
  *** end of common block ***

IDPROM image block #1:

  block-signature = 0x6003, block-version = 2,
  block-length = 103, block-checksum = 2556

  *** linecard specific block ***
  feature-bits =   00000000 00000000
  hardware-changes-bits =   00000000 00000000
  card index = 158
  mac base = 0012.4310.D840
  mac_len = 128
  num_processors = 1
  epld_num = 0
  epld_versions = 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000
  port numbers:
    pair #0: type=00, count=00
    pair #1: type=00, count=00
    pair #2: type=00, count=00
    pair #3: type=00, count=00
    pair #4: type=00, count=00
    pair #5: type=00, count=00
    pair #6: type=00, count=00
    pair #7: type=00, count=00
  sram_size = 0
  sensor_thresholds =
    sensor #0: critical = 75 oC, warning = 60 oC
    sensor #1: critical = 70 oC, warning = 55 oC
    sensor #2: critical = 80 oC, warning = 65 oC
    sensor #3: critical = 75 oC, warning = 60 oC
    sensor #4: critical = -128 oC (sensor not present), warning = -128 oC (sensor not
present)
    sensor #5: critical = -128 oC (sensor not present), warning = -128 oC (sensor not
present)
    sensor #6: critical = -128 oC (sensor not present), warning = -128 oC (sensor not
present)
    sensor #7: critical = -128 oC (sensor not present), warning = -128 oC (sensor not
present)
  max_connector_power = 3600
  cooling_requirement = 35
  ambient_temp = 55
  *** end of linecard specific block ***

End of IDPROM image
```

The following example shows sample output for a 4-Port OC-3c/STM-1 ATM SPA installed in subslot 0 of the SIP installed in slot 5 of the router:

```
Router# show idprom module 5/0

IDPROM for SPA module #5/0
        (FRU is '4-port OC3/STM1 ATM Shared Port Adapter')
        Product Identifier (PID) : SPA-4XOC3-ATM
        Version Identifier (VID) : V01
        PCB Serial Number        : PRTA2604138
```

```
        Top Assy. Part Number    : 68-2177-01
        73/68 Board Revision     : 05
        73/68 Board Revision     : 01
        Hardware Revision        : 0.224
        CLEI Code                : UNASSIGNED
```

The following example shows sample output for the **clei** form of the command for a 4-Port OC-3c/STM-1 POS SPA installed in subslot 3 of the SIP installed in slot 2 of the router:

```
Router# show idprom module 2/3 clei

FRU             PID                  VID SN          CLEI
--------------- -------------------- --- ----------- ----------
SPA module #2/3 SPA-4XOC3-POS        V01 PRTA0304155 UNASSIGNED
```

The following example shows sample output for the **detail** form of the command for a 4-Port OC-3c/STM-1 POS SPA installed in subslot 3 of the SIP installed in slot 2 of the router:

```
Router# show idprom module 2/3 detail

IDPROM for SPA module #2/3
        (FRU is '4-port OC3/STM1 POS Shared Port Adapter')
        EEPROM version           : 4
        Compatible Type          : 0xFF
        Controller Type          : 1088
        Hardware Revision        : 0.230
        Boot Timeout             : 0 msecs
        PCB Serial Number        : PRTA0304155
        Part Number              : 73-9313-02
        73/68 Board Revision     : 04
        Fab Version              : 02
        RMA Test History         : 00
        RMA Number               : 0-0-0-0
        RMA History              : 00
        Deviation Number         : 0
        Product Identifier (PID) : SPA-4XOC3-POS
        Version Identifier (VID) : V01
        Top Assy. Part Number    : 68-2169-01
        73/68 Board Revision     : 10
        System Clock Frequency   : 00 00 00 00 00 00 00 00
                                   00 00 00 00 00
        CLEI Code                : UNASSIGNED
        Base MAC Address         : 00 00 00 00 00 00
        MAC Address block size   : 0
        Manufacturing Test Data  : 00 00 00 00 00 00 00 00
        Field Diagnostics Data   : 00 00 00 00 00 00 00 00
        Calibration Data         : Minimum: 0 dBmV, Maximum: 0 dBmV
              Calibration values :
        Power Consumption        : 16200 mWatts (Maximum)
        Environment Monitor Data : 01 08 F6 48 43 34 F6 48
                                   43 34 02 31 0C E4 46 32
                                   28 13 07 09 C4 46 32 28
                                   13 07 00 00 00 00 00 00
                                   00 05 DC 46 32 28 13 07
                                   00 00 00 00 00 00 00 00
                                   00 00 00 00 00 00 00 00
                                   00 00 00 00 00 FE 02 00
                                   00
        Asset ID                 :
        Asset Alias              :
```

show inventory

To display the product inventory listing of all Cisco products installed in the networking device, use the **show inventory** command in user EXEC or privileged EXEC mode.

show inventory [raw] [*entity***]**

Syntax Description

raw	(Optional) Retrieves information about all of the Cisco products—referred to as entities—installed in the Cisco networking device, even if the entities do not have a product ID (PID) value, a unique device identifier (UDI), or other physical identification.
entity	(Optional) Name of a Cisco entity (for example, chassis, backplane, module, or slot). A quoted string may be used to display very specific UDI information; for example "sfslot 1" will display the UDI information for slot 1 of an entity named sfslot.

Command Modes

User EXEC
Privileged EXEC

Command History

Release	Modification
12.3(4)T	This command was introduced.
12.0(27)S	This command was integrated into Cisco IOS Release 12.0(27)S.
12.2(25)S	This command was integrated into Cisco IOS Release 12.2(25)S.
12.2(27)SBC	This command was integrated into Cisco IOS Release 12.2(27)SBC.
12.2(18)SXE5	This command was integrated into Cisco IOS Release 12.2(18)SXE5.

Usage Guidelines

The **show inventory** command retrieves and displays inventory information about each Cisco product in the form of a UDI. The UDI is a combination of three separate data elements: a product identifier (PID), a version identifier (VID), and the serial number (SN).

The PID is the name by which the product can be ordered; it has been historically called the "Product Name" or "Part Number." This is the identifier that one would use to order an exact replacement part.

The VID is the version of the product. Whenever a product has been revised, the VID will be incremented. The VID is incremented according to a rigorous process derived from Telcordia GR-209-CORE, an industry guideline that governs product change notices.

The SN is the vendor-unique serialization of the product. Each manufactured product will carry a unique serial number assigned at the factory, which cannot be changed in the field. This is the means by which to identify an individual, specific instance of a product.

The UDI refers to each product as an entity. Some entities, such as a chassis, will have subentities like slots. Each entity will display on a separate line in a logically ordered presentation that is arranged hierarchically by Cisco entities.

Use the **show inventory** command without options to display a list of Cisco entities installed in the networking device that are assigned a PID.

Examples

The following is sample output from the **show inventory** command without any keywords or arguments. This sample output displays a list of Cisco entities installed in a router that are assigned a PID.

```
Router# show inventory

NAME: "Chassis", DESCR: "12008/GRP chassis"
PID: GSR8/40           ,  VID: V01,  SN: 63915640

NAME: "slot 0", DESCR: "GRP"
PID: GRP-B             ,  VID: V01,  SN: CAB021300R5

NAME: "slot 1", DESCR: "4 port ATM OC3 multimode"
PID: 4OC3/ATM-MM-SC    ,  VID: V01,  SN: CAB04036GT1

NAME: "slot 3", DESCR: "4 port 0C3 POS multimode"
PID: LC-4OC3/POS-MM    ,  VID: V01,  SN: CAB014900GU

NAME: "slot 5", DESCR: "1 port Gigabit Ethernet"
PID: GE-GBIC-SC-B      ,  VID: V01,  SN: CAB034251NX

NAME: "slot 7", DESCR: "GRP"
PID: GRP-B             ,  VID: V01,  SN: CAB0428AN4O

NAME: "slot 16", DESCR: "GSR 12008 Clock Scheduler Card"
PID: GSR8-CSC/ALRM     ,  VID: V01,  SN: CAB0429AUYH

NAME: "sfslot 1", DESCR: "GSR 12008 Switch Fabric Card"
PID: GSR8-SFC          ,  VID: V01,  SN: CAB0428ALOS

NAME: "sfslot 2", DESCR: "GSR 12008 Switch Fabric Card"
PID: GSR8-SFC          ,  VID: V01,  SN: CAB0429AU0M

NAME: "sfslot 3", DESCR: "GSR 12008 Switch Fabric Card"
PID: GSR8-SFC          ,  VID: V01,  SN: CAB0429ARD7

NAME: "PSslot 1", DESCR: "GSR 12008 AC Power Supply"
PID: FWR-GSR8-AC-B     ,  VID: V01,  SN: CAB041999CW
```

Table 89 describes the fields shown in the display.

Table 89 show inventory Field Descriptions

Field	Description
NAME	Physical name (text string) assigned to the Cisco entity. For example, console or a simple component number (port or module number), such as "1," depending on the physical component naming syntax of the device.
DESCR	Physical description of the Cisco entity that characterizes the object. The physical description includes the hardware serial number and the hardware revision.
PID	Entity product identifier. Equivalent to the entPhysicalModelName MIB variable in RFC 2737.
VID	Entity version identifier. Equivalent to the entPhysicalHardwareRev MIB variable in RFC 2737.
SN	Entity serial number. Equivalent to the entPhysicalSerialNum MIB variable in RFC 2737.

For diagnostic purposes, the **show inventory** command can be used with the **raw** keyword to display every RFC 2737 entity including those without a PID, UDI, or other physical identification.

Note The **raw** keyword option is primarily intended for troubleshooting problems with the **show inventory** command itself.

```
Router# show inventory raw

NAME: "Chassis", DESCR: "12008/GRP chassis"
PID:                   ,  VID: V01,  SN: 63915640

NAME: "slot 0", DESCR: "GRP"
PID:                   ,  VID: V01,  SN: CAB021300R5

NAME: "slot 1", DESCR: "4 port ATM OC3 multimode"
PID: 4OC3/ATM-MM-SC    ,  VID: V01,  SN: CAB04036GT1

NAME: "slot 3", DESCR: "4 port 0C3 POS multimode"
PID: LC-4OC3/POS-MM    ,  VID: V01,  SN: CAB014900GU
```

Enter the **show inventory** command with an *entity* argument value to display the UDI information for a specific type of Cisco entity installed in the networking device. In this example, a list of Cisco entities that match the sfslot argument string is displayed.

```
Router# show inventory sfslot

NAME: "sfslot 1", DESCR: "GSR 12008 Switch Fabric Card"
PID: GSR8-SFC          ,  VID: V01,  SN: CAB0428ALOS

NAME: "sfslot 2", DESCR: "GSR 12008 Switch Fabric Card"
PID: GSR8-SFC          ,  VID: V01,  SN: CAB0429AU0M

NAME: "sfslot 3", DESCR: "GSR 12008 Switch Fabric Card"
PID: GSR8-SFC          ,  VID: V01,  SN: CAB0429ARD7
```

You can request even more specific UDI information using the **show inventory** command with an *entity* argument value that is enclosed in quotation marks. In this example, only the details for the entity that exactly matches the sfslot 1 argument string are displayed.

```
Router# show inventory "sfslot 1"

NAME: "sfslot 1", DESCR: "GSR 12008 Switch Fabric Card"
PID: GSR8-SFC          ,  VID: V01,  SN: CAB0428ALOS
```

Related Commands

Command	Description
show diag	Displays diagnostic information about the controller, interface processor, and port adapters for a networking device.
show tech-support	Displays general information about the router when it reports a problem.

show logging

To display the state of system logging (syslog) and the contents of the standard system logging buffer, use the **show logging** command in privileged EXEC mode.

show logging [**slot** *slot-number* | **summary**]

Syntax Description

slot *slot-number*	(Optional) Displays information in the syslog history table for a specific line card. Slot numbers range from 0 to 11 for the Cisco 12012 Internet router and 0 to 7 for the Cisco 12008 Internet router.
summary	(Optional) Displays counts of messages by type for each line card.

Command Modes

Privileged EXEC

Command History

Release	Modification
10.0	This command was introduced.
11.2 GS	The **slot** and **summary** keywords were added for the Cisco 12000.
12.2(8)T	Command output was expanded to show the status of the logging count facility (“Count and timestamp logging messages”).
12.2(15)T	Command output was expanded to show the status of XML syslog formatting.
12.3(2)T	Command output was expanded (on supported software images) to show details about the status of system logging processed through the Embedded Syslog Manager (ESM). These lines appear as references to “filtering” or “filter modules”.
12.3(2)XE	This command was integrated into Cisco IOS Release 12.3(2)XE.
12.2(25)S	This command was integrated into Cisco IOS Release 12.2(25)S.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.4(11)T	Command-line interface (CLI) output was modified to show message discriminators defined at the router and syslog sessions associated with those message discriminators.

Usage Guidelines

This command displays the state of syslog error and event logging, including host addresses, and which logging destinations (console, monitor, buffer, or host) logging is enabled. This command also displays Simple Network Management Protocol (SNMP) logging configuration parameters and protocol activity.

This command will also display the contents of the standard system logging buffer, if logging to the buffer is enabled. Logging to the buffer is enabled or disabled using the [**no**] **logging buffered** command. The number of system error and debugging messages in the system logging buffer is determined by the configured size of the syslog buffer. This size of the syslog buffer is also set using the **logging buffered** command.

To enable and set the format for syslog message timestamping, use the **service timestamps log** command.

If debugging is enabled (using any **debug** command), and the logging buffer is configured to include level 7 (debugging) messages, debug output will be included in the system log. Debugging output is not formatted like system error messages and will not be preceded by the percent symbol (%).

Examples

The following is sample output from the **show logging** command on a software image that supports the Embedded Syslog Manager (ESM) feature:

```
Router# show logging

Syslog logging: enabled (10 messages dropped, 5 messages rate-limited,
                0 flushes, 0 overruns, xml disabled, filtering disabled)
    Console logging: level debugging, 31 messages logged, xml disabled,
                     filtering disabled
    Monitor logging: disabled
    Buffer logging: level errors, 36 messages logged, xml disabled,
                    filtering disabled
    Logging Exception size (8192 bytes)
    Count and timestamp logging messages: disabled

No active filter modules.


    Trap logging: level informational, 45 message lines logged

Log Buffer (8192 bytes):
```

The following example shows output from the **show logging** command after a message discriminator has been configured. Included in this example is the command to configure the message discriminator.

```
c7200-3(config)# logging discriminator ATTFLTR1 severity includes 1,2,5 rate-limit 100

Specified MD by the name ATTFLTR1 is not found.
Adding new MD instance with specified MD attribute values.

Router(config)# end
Router#

000036: *Oct 20 16:26:04.570: %SYS-5-CONFIG_I: Configured from console by console

Router# show logging

Syslog logging: enabled (11 messages dropped, 0 messages rate-limited,
    0 flushes, 0 overruns, xml disabled, filtering disabled)

No Active Message Discriminator.

Inactive Message Discriminator:
ATTFLTR1  severity group includes 1,2,5
    rate-limit not to exceed 100 messages per second

Console logging: level debugging, 25 messages logged, xml disabled, filtering disabled
Monitor logging: level debugging, 0 messages logged, xml disabled, filtering disabled
Buffer logging:  level debugging, 25 messages logged, xml disabled, filtering disabled
Logging Exception size (8192 bytes)
Count and timestamp logging messages: disabled

No active filter modules.

Trap logging: level debugging, 28 message lines logged
Logging to 172.25.126.15  (udp port 1300,  audit disabled, authentication disabled,
    encryption disabled, link up),
```

```
    28 message lines logged,
    0 message lines rate-limited,
    0 message lines dropped-by-MD,
    xml disabled, sequence number disabled
    filtering disabled
Logging to 172.25.126.15  (tcp port 1307,  audit disabled, authentication disabled,
    encryption disabled, link up),
    28 message lines logged,
    0 message lines rate-limited,
    0 message lines dropped-by-MD,
    xml disabled, sequence number disabled, filtering disabled
Logging to 172.20.1.1  (udp port 514,  audit disabled,
    authentication disabled, encryption disabled, link up),
    28 message lines logged,
    0 message lines rate-limited,
    0 message lines dropped-by-MD,
    xml disabled, sequence number disabled
    filtering disabled

Log Buffer (1000000 bytes):
```

Table 90 describes the significant fields shown in the output for the two preceding examples.

Table 90 show logging Field Descriptions

Field	Description
Syslog logging:	Shows general state of system logging (enabled or disabled), the status of logged messages (number of messages dropped, rate-limited, or flushed), and whether XML formatting or ESM filtering is enabled.
No Active Message Discriminator	Indicates that a message discriminator is not being used.
Inactive Message Discriminator:	Identifies a configured message discriminator that has not been invoked.
Console logging:	Logging to the console port. Shows “disabled” or, if enabled, the severity level limit, number of messages logged, and whether XML formatting or ESM filtering is enabled. Corresponds to the configuration of the **logging console**, **logging console xml**, or **logging console filtered** command.
Monitor logging:	Logging to the monitor (all TTY lines). Shows “disabled” or, if enabled, the severity level limit, number of messages logged, and whether XML formatting or ESM filtering is enabled. Corresponds to the configuration of the **logging monitor**, **logging monitor xml**, or **logging monitor filtered** command.
Buffer logging:	Logging to the standard syslog buffer. Shows “disabled” or, if enabled, the severity level limit, number of messages logged, and whether XML formatting or ESM filtering is enabled. Corresponds to the configuration of the **logging buffered**, **logging buffered xml**, or **logging buffered filtered** command.

Table 90 show logging Field Descriptions (continued)

Field	Description
Trap logging:	Logging to a remote host (syslog collector). Shows "disabled" or, if enabled, the severity level limit, number of messages logged, and whether XML formatting or ESM filtering is enabled. (The word "trap" means a trigger in the system software for sending error messages to a remote host.) Corresponds to the configuration of the **logging host** command. The severity level limit is set using the **logging trap** command.
SNMP logging	Displays whether SNMP logging is enabled, the number of messages logged, and the retransmission interval. If not shown on your platform, use the **show logging history** command.
Logging Exception size (8192 bytes)	Corresponds to the configuration of the **logging exception** command.
Count and timestamp logging messages:	Corresponds to the configuration of the **logging count** command.
No active filter modules.	Appears if no syslog filter modules are configured with the **logging filter** command. Syslog filter modules are Tcl script files used when the Embedded Syslog Manager (ESM) is enabled. ESM is enabled when any of the **filtered** keywords are used in the logging commands. If configured, the URL and filename of configured syslog filter modules will appear at this position in the output. Syslog filter modules are executed in the order in which they appear here.
Log Buffer (8192 bytes):	The value in parentheses corresponds to the configuration of the **logging buffered** *buffer-size* command. If no messages are currently in the buffer, the output ends with this line. If messages are stored in the syslog buffer, they appear after this line.

The following example shows that syslog messages from the system buffer are included, with time stamps. In this example, the software image does not support XML formatting or ESM filtering of syslog messages.

```
Router# show logging

Syslog logging:enabled (2 messages dropped, 0 flushes, 0 overruns)
    Console logging:disabled
    Monitor logging:level debugging, 0 messages logged
    Buffer logging:level debugging, 4104 messages logged
    Trap logging:level debugging, 4119 message lines logged
        Logging to 192.168.111.14, 4119 message lines logged
Log Buffer (262144 bytes):

Jul 11 12:17:49 EDT:%BGP-4-MAXPFX:No. of prefix received from 209.165.200.225
(afi 0) reaches 24, max 24
! THE FOLLOWING LINE IS A DEBUG MESSAGE FROM NTP.
! NOTE THAT IT IS NOT PRECEEDED BY THE % SYMBOL.
Jul 11 12:17:48 EDT: NTP: Maxslew = 213866
Jul 11 15:15:41 EDT:%SYS-5-CONFIG:Configured from
tftp://host.com/addc5505-rsm.nyiix
.Jul 11 15:30:28 EDT:%BGP-5-ADJCHANGE:neighbor 209.165.200.226 Up
```

```
.Jul 11 15:31:34 EDT:%BGP-3-MAXPFXEXCEED:No. of prefix received from
209.165.200.226 (afi 0):16444 exceed limit 375
.Jul 11 15:31:34 EDT:%BGP-5-ADJCHANGE:neighbor 209.165.200.226 Down BGP
Notification sent
.Jul 11 15:31:34 EDT:%BGP-3-NOTIFICATION:sent to neighbor 209.165.200.226 3/1
(update malformed) 0 bytes
 .
 .
 .
```

The software clock keeps an "authoritative" flag that indicates whether the time is authoritative (believed to be accurate). If the software clock has been set by a timing source (for example, via NTP), the flag is set. If the time is not authoritative, it will be used only for display purposes. Until the clock is authoritative and the "authoritative" flag is set, the flag prevents peers from synchronizing to the software clock.

Table 91 describes the symbols that precede the timestamp.

Table 91 Timestamping Symbols for syslog Messages

Symbol	Description	Example
*	Time is not authoritative: the software clock is not in sync or has never been set.	*15:29:03.158 UTC Tue Feb 25 2003:
(blank)	Time is authoritative: the software clock is in sync or has just been set manually.	15:29:03.158 UTC Tue Feb 25 2003:
.	Time is authoritative, but NTP is not synchronized: the software clock was in sync, but has since lost contact with all configured NTP servers.	.15:29:03.158 UTC Tue Feb 25 2003:

The following is sample output from the **show logging summary** command for a Cisco 12012 router. A number in the column indicates that the syslog contains that many messages for the line card. For example, the line card in slot 9 has 1 error message, 4 warning messages, and 47 notification messages.

Note For similar log counting on other platforms, use the **show logging count** command.

```
Router# show logging summary

+-----+-------+-------+-------+-------+-------+-------+-------+-------+
 SLOT | EMERG | ALERT | CRIT  | ERROR |WARNING| NOTICE| INFO  | DEBUG |
+-----+-------+-------+-------+-------+-------+-------+-------+-------+
|* 0* |     . |     . |     . |     . |     . |     . |     . |     . |
|  1  |       |       |       |       |       |       |       |       |
|  2  |       |       |       |     1 |     4 |    45 |       |       |
|  3  |       |       |       |       |       |       |       |       |
|  4  |       |       |       |     5 |     4 |    54 |       |       |
|  5  |       |       |       |       |       |       |       |       |
|  6  |       |       |       |       |       |       |       |       |
|  7  |       |       |       |    17 |     4 |    48 |       |       |
|  8  |       |       |       |       |       |       |       |       |
|  9  |       |       |       |     1 |     4 |    47 |       |       |
| 10  |       |       |       |       |       |       |       |       |
| 11  |       |       |       |    12 |     4 |    65 |       |       |
+-----+-------+-------+-------+-------+-------+-------+-------+-------+
Router#
```

Table 92 describes the logging level fields shown in the display.

Table 92 show logging summary Field Descriptions

Field	Description
SLOT	Indicates the slot number of the line card. An asterisk next to the slot number indicates the GRP card whose error message counts are not displayed. For information on the GRP card, use the **show logging** command.
EMERG	Indicates that the system is unusable.
ALERT	Indicates that immediate action is needed.
CRIT	Indicates a critical condition.
ERROR	Indicates an error condition.
WARNING	Indicates a warning condition.
NOTICE	Indicates a normal but significant condition.
INFO	Indicates an informational message only.
DEBUG	Indicates a debugging message.

Related Commands

Command	Description
clear logging	Clears messages from the logging buffer.
logging count	Enables the error log count capability.
logging history size	Changes the number of syslog messages stored in the history table of the router.
logging linecard	Logs messages to an internal buffer on a line card and limits the logging messages displayed on terminal lines other than the console line to messages with a level at or above level.
service timestamps	Configures the system to timestamp debugging or logging messages.
show logging count	Displays a summary of system error messages (syslog messages) by facility and severity.
show logging xml	Displays the state of system logging and the contents of the XML-specific logging buffer.

show logging count

To display a summary of the number of times certain system error messages are occuring, use the **show logging** command in privileged EXEC mode.

show logging count

Syntax Description

This command has no arguements or keywords.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(8)T	This command was introduced.

Usage Guidelines

To enable the error log count capability (syslog counting feature), use the **logging count** command in global configuration mode.

This feature works independently of the various settings of the other logging commands (such as [**no**] **logging on**, [**no**] **logging buffered**, and so on). In other words, turning off logging by other means does not stop the counting and timestamping from occuring.

This command displays information such as the number of times a particular system error message occurs and the time stamp of the last occurrence of the specified message. System error messages are grouped into logical units called "Facilities" based on Cisco IOS software components.

To determine if system error message counting is enabled, use the **show logging** command.

The **service timestamps** command configuration determines the timestamp format (shown in the "Last Time" column) of **show logging count** command output. There is not quite enough space for all options of the possible options (datetime, milliseconds, and timezone) of the **service timestamps datetime** command to be displayed at the same time. As a result, if **msec** is selected, **timezone** will not be displayed. If **show-timezone** is selected but not **msec**, then the time zone will be displayed.

Occasionally, the length of the message name plus the facility name contains too many characters to be printed on one line. The CLI attempts to keep the name and facility name on one line but, if necessary, the line will be wrapped, so that the first line contains the facility name and the second line contains the message name and the rest of the columns.

Examples

The following example shows the number of times syslog messages have occurred and the most recent time that each error message occurred. In this example, the **show logging** command is used to determine if the syslog counting feature is enabled:

```
Router# show logging | include count
Count and timestamp logging messages: enabled

Router# show logging count

Facility       Message Name                    Sev  Occur  Last Time
===============================================================================
```

```
SYS             BOOTTIME                        6    1   00:00:12
SYS             RESTART                         5    1   00:00:11
SYS             CONFIG_I                        5    1   00:00:05
-------------   -----------------------------  ------------------------------
SYS TOTAL                                           3

LINEPROTO       UPDOWN                          5   13   00:00:19
-------------   -----------------------------  ------------------------------
LINEPROTO TOTAL                                    13

LINK            UPDOWN                          3    1   00:00:18
LINK            CHANGED                         5   12   00:00:09
-------------   -----------------------------  ------------------------------
LINK TOTAL                                         13

SNMP            COLDSTART                       5    1   00:00:11
-------------   -----------------------------  ------------------------------
SNMP TOTAL                                          1
```

Table 93 describes the significant fields shown in the display.

Table 93 show logging count Field Descriptions

Field	Description
Facility	The facility, such as syslog, from which these error messages are occurring.
Message Name	The name of this message.
Sev	The severity level of this message.
Occur	How many times this message has occurred.
Last Time	The last (most recent) time this message occurred. Timestamping is by default based on the system uptime (for example “3w1d” indicates 3 weeks and 1 day from the last system reboot.)
Sys Total / Lineproto Total / Link Total / SNMP Total	Total number of error messages that have occurred for the specified Facility.

In the following example, the user is interested only in the totals:

```
Router# show logging count | include total
SYS TOTAL                                           3
LINEPROTO TOTAL                                    13
LINK TOTAL                                         13
SNMP TOTAL                                          1
```

Related Commands

Command	Description
clear logging	Clears messages from the logging buffer.
logging count	Enables the system error message log count capability.
service timestamps	Configures the system to time-stamp debugging or logging messages.
show logging	Displays general information about the state of system logging.

show logging history

To display information about the state of the syslog history table, use the **show logging history** command in privileged EXEC mode.

show logging history

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Command History

Release	Modification
10.0	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

This command displays information about the syslog history table, such as the table size, the status of messages, and text of messages stored in the table. Messages stored in the table are governed by the **logging history** global configuration command.

Examples

The following example shows sample output from the **show logging history** command. In this example, notifications of severity level 5 (notifications) through severity level 0 (emergencies) are configured to be written to the logging history table.

```
Router# show logging history

Syslog History Table: 1 maximum table entries,
saving level notifications or higher
0 messages ignored, 0 dropped, 15 table entries flushed,
SNMP notifications not enabled
  entry number 16: SYS-5-CONFIG_I
  Configured from console by console
  timestamp: 1110
Router#
```

Table 94 describes the significant fields shown in the output.

Table 94 show logging history Field Descriptions

Field	Description
maximum table entry	Number of messages that can be stored in the history table. Set with the **logging history size** command.
saving level notifications <x> or higher	Level of messages that are stored in the history table and sent to the SNMP server (if SNMP notification is enabled). The severity level can be configured with the **logging history** command.

Table 94 show logging history Field Descriptions (continued)

Field	Description
messages ignored	Number of messages not stored in the history table because the severity level is greater than that specified with the **logging history** command.
dropped	Number of messages that could not be processed due to lack of system resources. Dropped messages do not appear in the history table and are not sent to the SNMP server.
table entries flushed	Number of messages that have been removed from the history table to make room for newer messages.
SNMP notifications	Whether syslog traps of the appropriate level are sent to the SNMP server. The sending of syslog traps are enabled or disabled through the **snmp-server enable traps syslog** command.
entry number:	Number of the message entry in the history table. In the example above, the message "SYS-5-CONFIG_I Configured from console by console" indicates a syslog message consisting of the facility name (SYS), which indicates where the message came from, the severity level (5) of the message, the message name (CONFIG_I), and the message text.
timestamp	Time, based on the up time of the router, that the message was generated.

Related Commands

Command	Description
clear logging	Clears messages from the logging buffer.
logging history	Limits syslog messages sent to the router's history table to a specified severity level.
logging history size	Changes the number of syslog messages that can be stored in the history table.
logging linecard	Logs messages to an internal buffer on a line card. This command limits the logging messages displayed on terminal lines other than the console line to messages with a level at or above level.
snmp-server enable traps	The [**no**] **snmp-server enable traps syslog** form of this command controls (enables or disables) the sending of system-logging messages to a network management station.

show logging system

To display the System Event Archive (SEA) logging system disk, use the **show logging system** command in privileged EXEC mode.

show logging system [**disk** | **size**]

Syntax Description

disk	(Optional) Displays the location of the SEA logging system disk.
size	(Optional) Displays the current size of the SEA.

Defaults

This command has no default settings.

Command Modes

Privileged EXEC

Usage Guidelines

The **show logging system** command displays the latest messages first.

Examples

The following example shows how to display the latest system log messages:

```
Router# show logging system

SEQ: MM/DD/YY HH:MM:SS MOD/SUB: SEV, COMP, MESSAGE
=================================================
1: 01/24/07 15:38:40 6/-1 : MAJ, GOLD, syndiagSyncPinnacle failed in slot 6
2: 01/24/07 15:38:40 6/-1 : MAJ, GOLD, queryHyperionSynched[6]: Hyperion out of sync in
sw_mode 1
3: 01/24/07 15:38:40 6/-1 : MAJ, GOLD, queryHyperionSynched[6]: Hyperion out of sync in
sw_mode 1
4: 01/24/07 15:38:40 6/-1 : MAJ, GOLD, queryHyperionSynched[6]: Hyperion out of sync in
sw_mode 1
5: 01/24/07 15:38:40 6/-1 : MAJ, GOLD, queryHyperionSynched[6]: Hyperion out of sync in
sw_mode 1
6: 01/24/07 15:38:40 6/-1 : MAJ, GOLD, queryHyperionSynched[6]: Hyperion out of sync in
sw_mode 1
7: 01/24/07 15:38:39 6/-1 : MAJ, GOLD, queryHyperionSynched[6]: Hyperion out of sync in
sw_mode 1
```

The following example shows how to display the SEA logging system disk:

```
Router# show logging system disk

SEA log disk: sup-bootdisk:
```

The following example shows how to display the current size of the SEA:

```
Router# show logging system size

SEA log size: 33554432 bytes
```

Table 95 describes the significant fields shown in the display.

Table 95 show logging system Field Descriptions

Field	Description
MOD/SUB	Indicates the source of the event message.
SEV	Indicates the severity level of the message.
COMP	Indicates the software component that has logged the message.

Related Commands

clear logging system	Clears the event records stored in the SEA.
copy logging system	Copies the archived system events to another device.
logging system	Enables or disables the System Event Archive logging.

show logging xml

To display the state of system message logging in an XML format, and to display the contents of the XML syslog buffer, use the **show logging xml** command in privileged EXEC mode.

show logging xml

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(15)T	This command was introduced.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.

Usage Guidelines

This command displays the same syslog state information as the standard **show logging** command, but displays the information in XML format. This command also displays the content of the XML syslog buffer (if XML-formatted buffer logging is enabled).

Examples

The following example compares the output of the standard **show logging** command with the output of the **show logging xml** command so that you can see how the standard information is formatted in XML.

```
Router# show logging

Syslog logging: enabled (10 messages dropped, 6 messages rate-limited, 0 flushes, 0
overruns, xml enabled)
    Console logging: level debugging, 28 messages logged, xml enabled
    Monitor logging: level debugging, 0 messages logged, xml enabled
    Buffer logging: level debugging, 2 messages logged, xml enabled (2 messages logged)
    Logging Exception size (8192 bytes)
    Count and timestamp logging messages: disabled
    Trap logging: level informational, 35 message lines logged
        Logging to 10.2.3.4, 1 message lines logged, xml disabled
        Logging to 192.168.2.1, 1 message lines logged, xml enabled

Log Buffer (8192 bytes):

00:04:20: %SYS-5-CONFIG_I: Configured from console by console
00:04:41: %SYS-5-CONFIG_I: Configured from console by console

Router# show logging xml

<syslog-logging status="enabled" msg-dropped="10" msg-rate-limited="6" flushes="0"
overruns="0"><xml>enabled</xml></syslog-logging>
    <console-logging level="debugging"
messages-logged="28"><xml>enabled</xml></console-logging>
    <monitor-logging level="debugging"
messages-logged="0"><xml>enabled</xml></monitor-logging>
    <buffer-logging level="debugging" messages-logged="2"><xml
messages-logged="2">enabled</xml></buffer-logging>
```

```
    <logging-exception size="8192 bytes"></logging-exception>
    <count-and-timestamp-logging status="disabled"></count-and-timestamp-logging>
    <trap-logging level="informational" messages-lines-logged="35"></trap-logging>
        <logging-to><dest id="0" ipaddr="10.2.3.4"
message-lines-logged="1"><xml>disabled</xml><dest></logging-to>
        <logging-to><dest id="1" ipaddr="192.168.2.1"
message-lines-logged="1"><xml>enabled</xml><dest></logging-to>

<log-xml-buffer size="44444 bytes"></log-xml-buffer>

<ios-log-msg><facility>SYS</facility><severity>5</severity><msg-id>CONFIG_I</msg-id><time>
00:04:20</time><args><arg id="0">console</arg><arg
id="1">console</arg></args></ios-log-msg>
<ios-log-msg><facility>SYS</facility><severity>5</severity><msg-id>CONFIG_I</msg-id><time>
00:04:41</time><args><arg id="0">console</arg><arg
id="1">console</arg></args></ios-log-msg>
Router#
```

Table 96 describes the significant fields shown in the displays.

Table 96 show logging and show logging xml Field Descriptions

Field	Description	XML Tag
Syslog logging	The global state of system message logging (syslog); “enabled” or “disabled.”	syslog-logging
Console logging	State of logging to console connections.	console-logging
Monitor logging	State of logging to monitor (TTY and Telnet) connections.	monitor-logging
Buffer logging	State of logging to the local system logging buffer.	buffer-logging
Count and timestamp logging messages:	Indicates whether the logging count feature is enabled. Corresponds to the **logging count** command.	count-and-timestamp-logging
Trap logging	State of logging to a remote host.	trap-logging

Related Commands

Command	Description
show logging	Displays the contents of the standard syslog buffer.
show logging count	Displays counts of each system error message.
show logging history	Displays the contents of the SNMP syslog history table.

show memory

To display statistics about memory when Cisco IOS or Cisco IOS software Modularity images are running, use the **show memory** command in user EXEC or privileged EXEC mode.

Cisco IOS Software

show memory [*memory-type*] [**free**] [**overflow**] [**summary**]

Cisco IOS Software Modularity

show memory

Syntax Description

memory-type	(Optional) Memory type to display (**processor**, **multibus**, **io**, or **sram**). If *memory-type* is not specified, statistics for all memory types present are displayed.
free	(Optional) Displays free memory statistics.
overflow	(Optional) Displays details about memory block header corruption corrections when the **exception memory ignore overflow** global configuration command is configured.
summary	(Optional) Displays a summary of memory usage including the size and number of blocks allocated for each address of the system call that allocated the block.

Command Modes

User EXEC (>)
Privileged EXEC (#)

Command History

Release	Modification
10.0	This command was introduced.
12.3(7)T	This command was enhanced with the **overflow** keyword to display details about memory block header corruption corrections.
12.2(25)S	The command output was updated to display information about transient memory pools.
12.3(14)T	The command output was updated to display information about transient memory pools.
12.2(27)SBC	This command was integrated into Cisco IOS Release 12.2(27)SBC.
12.2(18)SXF4	This command was implemented in Cisco IOS Software Modularity images.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

Cisco IOS Software

The **show memory** command displays information about memory available after the system image decompresses and loads.

Cisco IOS Software Modularity

No optional keywords or arguments are supported for the **show memory** command when a Software Modularity image is running. To display details about PSOIX and Cisco IOS style system memory information when Software Modularity images are running, use the **show memory detailed** command.

Examples

Example output varies between Cisco IOS software images and Cisco IOS Software Modularity software images. To view the appropriate output, choose one of the following sections:

- Cisco IOS Software
- Cisco IOS Software Modularity

Cisco IOS Software

The following is sample output from the **show memory** command:

```
Router# show memory

              Head   Total(b)    Used(b)    Free(b)  Lowest(b) Largest(b)
Processor    B0EE38    5181896    2210036    2971860    2692456    2845368

          Processor memory
Address   Bytes Prev.    Next     Ref  PrevF   NextF   Alloc PC  What
B0EE38     1056 0         B0F280     1                 18F132    List Elements
B0F280     2656 B0EE38    B0FD08     1                 18F132    List Headers
B0FD08     2520 B0F280    B10708     1                 141384    TTY data
B10708     2000 B0FD08    B10F00     1                 14353C    TTY Input Buf
B10F00      512 B10708    B11128     1                 14356C    TTY Output Buf
B11128     2000 B10F00    B11920     1                 1A110E    Interrupt Stack
B11920       44 B11128    B11974     1                 970DE8    *Init*
B11974     1056 B11920    B11DBC     1                 18F132    messages
B11DBC       84 B11974    B11E38     1                 19ABCE    Watched Boolean
B11E38       84 B11DBC    B11EB4     1                 19ABCE    Watched Boolean
B11EB4       84 B11E38    B11F30     1                 19ABCE    Watched Boolean
B11F30       84 B11EB4    B11FAC     1                 19ABCE    Watched Boolean
```

The following is sample output from the **show memory free** command:

```
Router# show memory free

              Head   Total(b)    Used(b)    Free(b)  Lowest(b) Largest(b)
Processor     B0EE38    5181896    2210076    2971820    2692456    2845368

          Processor memory
Address   Bytes Prev.    Next     Ref  PrevF   NextF   Alloc PC  What
             24    Free list 1
CEB844       32  CEB7A4 CEB88C      0  0       0       96B894    SSE Manager
             52    Free list 2
             72    Free list 3
             76    Free list 4
             80    Free list 5
D35ED4       80 D35E30   D35F4C     0  0       D27AE8  96B894    SSE Manager
D27AE8       80 D27A48   D27B60     0  D35ED4  0       22585E    SSE Manager
             88    Free list 6
            100    Free list 7
D0A8F4      100 D0A8B0   D0A980     0  0       0       2258DA    SSE Manager
            104    Free list 8
B59EF0      108 B59E8C   B59F84     0  0       0       2258DA    (fragment)
```

The output of the **show memory free** command contains the same types of information as the **show memory** output, except that only free memory is displayed, and the information is ordered by free list.

The first section of the display includes summary statistics about the activities of the system memory allocator. Table 97 describes the significant fields shown in the first section of the display.

Table 97 show memory Field Descriptions—First Section

Field	Description
Head	Hexadecimal address of the head of the memory allocation chain.
Total(b)	Sum of used bytes plus free bytes.
Used(b)	Amount of memory in use.
Free(b)	Amount of memory not in use.
Lowest(b)	Smallest amount of free memory since last boot.
Largest(b)	Size of largest available free block.

The second section of the display is a block-by-block listing of memory use. Table 98 describes the significant fields shown in the second section of the display.

Table 98 Characteristics of Each Block of Memory—Second Section

Field	Description
Address	Hexadecimal address of block.
Bytes	Size of block (in bytes).
Prev.	Address of previous block (should match the address on previous line).
Next	Address of next block (should match the address on next line).
Ref	Reference count for that memory block, indicating how many different processes are using that block of memory.
PrevF	Address of previous free block (if free).
NextF	Address of next free block (if free).
Alloc PC	Address of the system call that allocated the block.
What	Name of process that owns the block, or “(fragment)” if the block is a fragment, or “(coalesced)” if the block was coalesced from adjacent free blocks.

The **show memory io** command displays the free I/O memory blocks. On the Cisco 4000 router, this command quickly shows how much unused I/O memory is available.

The following is sample output from the **show memory io** command:

```
Router# show memory io

Address   Bytes Prev.   Next     Ref  PrevF   NextF   Alloc PC  What
6132DA0   59264 6132664 6141520  0    0       600DDEC  3FCF0     *Packet Buffer*
600DDEC     500 600DA4C 600DFE0  0   6132DA0 600FE68  0
600FE68     376 600FAC8 600FFE0  0   600DDEC 6011D54  0
6011D54     652 60119B4 6011FEO  0   600FE68 6013D54  0
614FCA0     832 614F564 614FFE0  0   601FD54 6177640  0
6177640 2657056 6172E90 0        0   614FCA0 0        0
Total: 2723244
```

The following example displays details of a memory block overflow correction when the **exception memory ignore overflow** global configuration command is configured:

```
Router# show memory overflow

Count    Buffer Count     Last corrected      Crashinfo files

1        1                00:11:17            slot0:crashinfo_20030620-075755
Traceback   607D526C 608731A0 607172F8 607288E0 607A5688 607A566C
```

The report includes the amount of time since the last correction was made and the name of the file that logged the memory block overflow details.

The **show memory sram** command displays the free SRAM memory blocks. For the Cisco 4000 router, this command supports the high-speed static RAM memory pool to make it easier for you to debug or diagnose problems with allocation or freeing of such memory.

The following is sample output from the **show memory sram** command:

```
Router# show memory sram

Address    Bytes Prev.    Next      Ref  PrevF    NextF    Alloc PC  What
7AE0       38178 72F0     0         0    0        0        0
Total      38178
```

The following example of the **show memory** command used on the Cisco 4000 router includes information about SRAM memory and I/O memory:

```
Router# show memory

               Head   Total(b)    Used(b)    Free(b)  Lowest(b) Largest(b)
Processor    49C724   28719324    1510864   27208460   26511644   15513908
      I/O   6000000    4194304    1297088    2897216    2869248    2896812
     SRAM      1000      65536      63400       2136       2136       2136

Address    Bytes Prev.    Next      Ref  PrevF    NextF    Alloc PC  What
1000        2032 0        17F0        1                    3E73E     *Init*
17F0        2032 1000     1FE0        1                    3E73E     *Init*
1FE0         544 17F0     2200        1                    3276A     *Init*
2200          52 1FE0     2234        1                    31D68     *Init*
2234          52 2200     2268        1                    31DAA     *Init*
2268          52 2234     229C        1                    31DF2     *Init*
72F0        2032 6E5C     7AE0        1                    3E73E     Init
7AE0       38178 72F0     0           0    0       0       0
```

The **show memory summary** command displays a summary of all memory pools and memory usage per Alloc PC (address of the system call that allocated the block).

The following is a partial sample output from the **show memory summary** command. This output shows the size, blocks, and bytes allocated. Bytes equal the size multiplied by the blocks. For a description of the other fields, see Table 97 and Table 98.

```
Router# show memory summary

Head   Total(b)    Used(b)    Free(b)  Lowest(b) Largest(b)
Processor    B0EE38    5181896    2210216    2971680    2692456    2845368

          Processor memory
Alloc PC        Size     Blocks      Bytes    What
0x2AB2           192          1        192    IDB: Serial Info
0x70EC            92          2        184    Init
0xC916           128         50       6400    RIF Cache
0x76ADE         4500          1       4500    XDI data
0x76E84         4464          1       4464    XDI data
```

```
0x76EAC          692          1          692    XDI data
0x77764          408          1          408    Init
0x77776          116          1          116    Init
0x777A2          408          1          408    Init
0x777B2          116          1          116    Init
0xA4600           24          3           72    List
0xD9B5C           52          1           52    SSE Manager
.
.
.
0x0                0       3413      2072576    Pool Summary
0x0                0         28      2971680    Pool Summary (Free Blocks)
0x0               40       3441       137640    Pool Summary (All Block Headers)
0x0                0       3413      2072576    Memory Summary
0x0                0         28      2971680    Memory Summary (Free Blocks)
```

Cisco IOS Software Modularity

The following is sample output from the **show memory** command when a Cisco IOS Software Modularity image is running.

```
Router# show memory

System Memory: 262144K total, 116148K used, 145996K free 4000K kernel reserved
```

Table 99 describes the significant fields shown in the display.

Table 99 show memory (Software Modularity Image) Field Descriptions

Field	Description
total	Total amount of memory on the device, in kilobytes.
used	Amount of memory in use, in kilobytes.
free	Amount of memory not in use, in kilobytes.
kernel reserved	Amount of memory reserved by the kernel, in kilobytes.

Related Commands

Command	Description
exception memory ignore overflow	Configures the Cisco IOS software to correct corruptions in memory block headers and allow a router to continue its normal operation.
show memory detailed	Displays POSIX and Cisco IOS style system memory information.
show processes memory	Displays memory used per process.

show memory allocating-process

To display statistics on allocated memory with corresponding allocating processes, use the **show memory allocating-process** command in user EXEC or privileged EXEC mode.

show memory allocating-process [**totals**]

Syntax Description

totals	(Optional) Displays allocating memory totals.

Command Modes

User EXEC
Privileged EXEC

Command History

Release	Modification
12.0	This command was introduced.

Usage Guidelines

The **show memory allocating-process** command displays information about memory available after the system image decompresses and loads.

Examples

The following is sample output from the **show memory allocating-process** command:

```
Router# show memory allocating-process

    Head Total(b)Used(b)Free(b)Lowest(b)Largest(b)
Processor 44E035601866326362613189616050074016040205215307820 4
     Fast 44DE35601310725828072792727927 9272764

          Processor memory

 Address  Bytes Prev.    Next     Ref    Alloc Proc    Alloc PC  What
6148EC40   1504 0        6148F24C   1  *Init*          602310FC  List Elements
6148F24C   3004 6148EC40 6148FE34   1  *Init*          60231128  List Headers
6148FE34   9000 6148F24C 61492188   1  *Init*          6023C634  Interrupt Stack
61492188     44 6148FE34 614921E0   1  *Init*          60C17FD8  *Init*
614921E0   9000 61492188 61494534   1  *Init*          6023C634  Interrupt Stack
61494534     44 614921E0 6149458C   1  *Init*          60C17FD8  *Init*
6149458C    220 61494534 61494694   1  *Init*          602450F4  *Init*
61494694   4024 6149458C 61495678   1  *Init*          601CBD64  TTY data
.
.
.
```

Table 100 describes the significant fields shown in the display.

Table 100 show memory allocating-process Field Descriptions

Field	Description
Head	Hexadecimal address of the head of the memory allocation chain.
Total(b)	Sum of used bytes plus free bytes.

Table 100 show memory allocating-process Field Descriptions (continued)

Field	Description
Used(b)	Amount of memory in use in bytes.
Free(b)	Amount of memory not in use (in bytes).
Lowest(b)	Smallest amount of free memory since last boot (in bytes).
Largest(b)	Size of largest available free block (in bytes).
Address	Hexadecimal address of the block.
Bytes	Size of the block (in bytes).
Prev.	Address of the preceding block (should match the address on preceding row).
Next	Address of the following block (should match the address on following row).
Ref	Reference count for that memory block, indicating how many different processes are using that block of memory.
Alloc PC	Address of the system call that allocated the block.
What	Name of process that owns the block, or "(fragment)" if the block is a fragment, or "(coalesced)" if the block was coalesced from adjacent free blocks.

The following is sample output from the **show memory allocating-process totals** command:

```
Router# show memory allocating-process totals

               Head    Total(b)     Used(b)     Free(b)   Lowest(b)  Largest(b)
Processor   44E03560   186632636    26142524   160490112   160402052   153078204
     Fast   44DE3560      131072       58280       72792       72792       72764

Allocator PC Summary for: Processor

    PC          Total   Count  Name
0x4041AF8C    5710616    3189  *Packet Data*
0x4041AF40    2845480    3190  *Packet Header*
0x404DBA28    1694556     203  Process Stack
0x4066EA68    1074080      56  Init
0x404B5F68    1049296       9  pak subblock chunk
0x41DCF230     523924      47  TCL Chunks
0x404E2488     448920       6  MallocLite
0x4066EA8C     402304      56  Init
0x40033878     397108       1  Init
0x41273E24     320052       1  CEF: table event ring
0x404B510C     253152      24  TW Buckets
0x42248F0C     229428       1  Init
0x42248F28     229428       1  Init
0x42248F48     229428       1  Init
0x423FF210     218048       5  Dn48oC!M
0x421CB530     208144       1  epa crypto blk
0x417A07F0     196764       3  L2TP Hash Table
0x403AFF50     187836       3  Init
```

Table 101 describes the significant fields shown in the display.

Table 101 show memory allocating-process totals Field Descriptions

Field	Description
Head	Hexadecimal address of the head of the memory allocation chain.
Total(b)	Sum of used bytes plus free bytes.
Used(b)	Amount of memory in use (in bytes).
Free(b)	Amount of memory not in use (in bytes).
Lowest(b)	Smallest amount of free memory since last boot (in bytes).
Largest(b)	Size of the largest available free block in bytes.
PC	Program counter
Total	Total memory allocated by the process (in bytes).
Count	Number of allocations.
Name	Name of the allocating process.

Related Commands

Command	Description
show processes memory	Displays memory used per process.

show memory dead

To display statistics on memory allocated by processes that have terminated, use the **show memory dead** command in user EXEC or privileged EXEC mode.

show memory dead [**totals**]

Syntax Description

totals	(Optional) Displays memory totals for processes that have been terminated.

Command Modes

User EXEC
Privileged EXEC

Command History

Release	Modification
12.0	This command was introduced.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

The **show memory dead** command displays information about processes that have been terminated. Terminated processes accounts for memory allocated under another process.

Examples

The following is sample output from the **show memory dead** command:

```
Router# show memory dead

              Head    Total(b) Used(b)     Free(b)   Lowest(b)  Largest(b)
      I/O    600000    2097152 461024    1636128    1635224    1635960

          Processor memory

 Address  Bytes Prev.    Next     Ref  PrevF   NextF   Alloc PC  What
1D8310       60 1D82C8   1D8378     1                  3281FFE   Router Init
2CA964       36 2CA914   2CA9B4     1                  3281FFE   Router Init
2CAA04      112 2CA9B4   2CAAA0     1                  3A42144   OSPF Stub LSA RBTree
2CAAA0       68 2CAA04   2CAB10     1                  3A420D4   Router Init
2ED714       52 2ED668   2ED774     1                  3381C84   Router Init
2F12AC       44 2F124C   2F1304     1                  3A50234   Router Init
2F1304       24 2F12AC   2F1348     1                  3A420D4   Router Init
2F1348       68 2F1304   2F13B8     1                  3381C84   Router Init
300C28      340 300A14   300DA8     1                  3381B42   Router Init
```

Table 102 describes the significant fields shown in the display.

Table 102 show memory dead Field Descriptions

Field	Description
Head	Hexadecimal address of the head of the memory allocation chain.
Total(b)	Sum of used bytes plus free bytes.
Used(b)	Amount of memory in use.
Free(b)	Amount of memory not in use (in bytes).
Lowest(b)	Smallest amount of free memory since last boot (in bytes).
Largest(b)	Size of the largest available free block (in bytes).
Address	Hexadecimal address of the block (in bytes).
Bytes	Size of the block (in bytes).
Prev.	Address of the preceding block.
Next	Address of the following block.
Ref	Reference count for that memory block, indicating how many different processes are using that block of memory.
PrevF	Address of the preceding free block (if free).
NextF	Address of the following free block (if free).
Alloc PC	Address of the program counter that allocated the block.
What	Name of the process that owns the block, or "(fragment)" if the block is a fragment, or "(coalesced)" if the block was coalesced from adjacent free blocks.

show memory debug incremental

To display information about memory leaks after a starting time has been established, use the **show memory debug incremental** command in privileged EXEC mode.

show memory debug incremental {**allocations** | **leaks** [**lowmem** | **summary**] | **status**}

Syntax Description

allocations	Displays all memory blocks that were allocated after issuing the **set memory debug incremental starting-time** command.
leaks	Displays only memory that was leaked after issuing the **set memory debug incremental starting-time** command.
lowmem	(Optional) Forces the memory leak detector to work in low memory mode, making no memory allocations.
summary	(Optional) Reports summarized memory leaks based on allocator_pc and size of the memory block.
status	Displays all memory blocks that were allocated after issuing the **set memory debug incremental starting-time** command.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.3(7)T	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.4T	The summary keyword was added.

Usage Guidelines

The **show memory debug incremental allocations** command displays all the memory blocks that were allocated after the **set memory debug incremental starting-time** command was entered. The displayed memory blocks are just memory allocations, they are not necessarily leaks.

The **show memory debug incremental leaks** command provides output similar to the **show memory debug leaks** command, except that it displays only memory that was leaked after the **set memory debug incremental starting-time** command was entered.

The **show memory debug incremental leaks lowmem** command forces memory leak detection to work in low memory mode. The amount of time taken for analysis is considerably greater than that of normal mode. The output for this command is similar to the **show memory debug leaks** command, except that it displays only memory that was leaked after the **set memory debug incremental starting-time** command was entered. You can use this command when you already know that normal mode memory leak detection will fail (perhaps by an unsuccessful previous attempt to invoke normal mode memory leak detection).

The **show memory debug incremental leaks summary** command displays a summarized report of the memory that was leaked after the **set memory debug incremental starting-time** command was entered, ordered by allocator process call address (Alloc_pc) and by memory block size.

The **show memory debug incremental status** command displays whether a starting point for incremental analysis has been set and the elapsed time since then.

Note All memory leak detection commands invoke normal mode memory leak detection, except when the low memory option is specifically invoked by use of the **lowmem** keyword. In normal mode, if memory leak detection determines that there is insufficient memory to proceed in normal mode, it will display an appropriate message and switch to low memory mode.

Examples

show memory debug incremental allocations Command Example

The following example shows output from the **show memory debug incremental** command when entered with the **allocations** keyword:

```
Router# show memory debug incremental allocations

Address    Size   Alloc_pc  PID  Name
62DA4E98      176 608CDC7C  44   CDP Protocol
62DA4F48       88 608CCCC8  44   CDP Protocol
62DA4FA0       88 606224A0  3    Exec
62DA4FF8       96 606224A0  3    Exec
635BF040       96 606224A0  3    Exec
63905E50      200 606A4DA4  69   Process Events
```

show memory debug incremental leaks summary Command Example

The following example shows output from the **show memory debug incremental** command when entered with the **leaks** and **summary** keywords:

```
Router# show memory debug incremental leaks summary
Adding blocks for GD...

                PCI memory
Alloc PC     Size     Blocks      Bytes    What
                I/O memory
Alloc PC     Size     Blocks      Bytes    What
                Processor memory
Alloc PC     Size          Blocks       Bytes      What
0x60874198  0000000052  0000000001  0000000052    Exec
0x60874198  0000000060  0000000001  0000000060    Exec
0x60874198  0000000100  0000000001  0000000100    Exec
0x60874228  0000000052  0000000004  0000000208    Exec
0x60874228  0000000060  0000000002  0000000120    Exec
0x60874228  0000000100  0000000004  0000000400   Exec
```

show memory debug incremental status Command Example

The following example shows output from the **show memory debug incremental** command entered with the **status** keyword:

```
Router# show memory debug incremental status

Incremental debugging is enabled
Time elapsed since start of incremental debugging: 00:00:10
```

Related Commands

Command	Description
set memory debug incremental starting-time	Sets the current time as the starting time for incremental analysis.
show memory debug leaks	Displays detected memory leaks.

show memory debug leaks

To display detected memory leaks, use the **show memory debug leaks** command in privileged EXEC mode.

show memory debug leaks [**chunks** | **largest** | **lowmem** | **summary**]

Syntax Description

chunks	(Optional) Displays the memory leaks in chunks.
largest	(Optional) Displays the top ten leaking allocator_pcs based on size, and the total amount of memory they have leaked.
lowmem	(Optional) Forces the memory leak detector to work in low memory mode, making no memory allocations.
summary	(Optional) Reports summarized memory leaks based on allocator_pc and size of the memory block.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.3(8)T1	This command was introduced.
12.2(25)S	This command was integrated into Cisco IOS Release 12.2(25)S.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

If no optional keywords are specified, the **show memory debug leaks** command invokes normal mode memory leak detection and does not look for memory leaks in chunks.

The **show memory debug leaks chunks** command invokes normal mode memory leak detection and looks for leaks in chunks as well.

The **show memory debug leaks largest** command displays the top ten leaking allocator_pcs and the total amount of memory that they have leaked. Additionally, each time this command is invoked it remembers the previous invocation's report and compares it to the current invocation's report. If there are new entries in the current report they are tagged as “inconclusive.” If the same entry appears in the previous invocation's report and the current invocation's report, the inconclusive tag is not added. It would be beneficial to run memory leak detection more than once and to consider only the consistently reported leaks.

The **show memory debug leaks lowmem** command forces memory leak detection to work in low memory mode. The amount of time taken for analysis is considerably greater than that of normal mode. The output for this command is similar to the **show memory debug leaks** command. You can use this command when you already know that normal mode memory leak detection will fail (perhaps by an unsuccessful previous attempt to invoke normal mode memory leak detection).

The **show memory debug leaks summary** command reports memory leaks based on allocator_pc and then on the size of the block.

Note All memory leak detection commands invoke normal mode memory leak detection, except when the low memory option is specifically invoked by use of the **lowmem** keyword. In normal mode, if memory leak detection determines that there is insufficient memory to proceed in normal mode, it will display an appropriate message and switch to low memory mode.

Examples

show memory debug leaks Command Example

The following example shows output from the **show memory debug leaks** command:

```
Router# show memory debug leaks

Adding blocks for GD...

                PCI memory
Address    Size   Alloc_pc  PID  Name

                I/O memory
Address    Size   Alloc_pc  PID  Name

                Processor memory
Address    Size   Alloc_pc  PID  Name
62DABD28     80 60616750  -2   Init
62DABD78     80 606167A0  -2   Init
62DCF240     88 605B7E70  -2   Init
62DCF298     96 605B7E98  -2   Init
62DCF2F8     88 605B7EB4  -2   Init
62DCF350     96 605B7EDC  -2   Init
63336C28    104 60C67D74  -2   Init
63370D58     96 60C656AC  -2   Init
633710A0    304 60C656AC  -2   Init
63B2BF68     96 60C659D4  -2   Init
63BA3FE0  32832 608D2848  104  Audit Process
63BB4020  32832 608D2FD8  104  Audit Process
```

Table 103 describes the significant fields shown in the display.

Table 103 show memory debug leaks Field Descriptions

Field	Description
Address	Hexadecimal address of the leaked block.
Size	Size of the leaked block (in bytes).
Alloc_pc	Address of the system call that allocated the block.
PID	The process identifier of the process that allocated the block.
Name	The name of the process that allocated the block.

show memory debug leaks chunks Command Example

The following example shows output from the **show memory debug leaks chunks** command:

```
Router# show memory debug leaks chunks

Adding blocks for GD...

                PCI memory
Address    Size   Alloc_pc  PID  Name
```

```
Chunk Elements:
Address  Size  Parent   Name

                I/O memory
Address    Size   Alloc_pc  PID  Name

Chunk Elements:
Address  Size  Parent   Name

                Processor memory
Address    Size   Alloc_pc  PID  Name
62DABD28       80 60616750  -2   Init
62DABD78       80 606167A0  -2   Init
62DCF240       88 605B7E70  -2   Init
62DCF298       96 605B7E98  -2   Init
62DCF2F8       88 605B7EB4  -2   Init
62DCF350       96 605B7EDC  -2   Init
63336C28      104 60C67D74  -2   Init
63370D58       96 60C656AC  -2   Init
633710A0      304 60C656AC  -2   Init
63B2BF68       96 60C659D4  -2   Init
63BA3FE0    32832 608D2848  104  Audit Process
63BB4020    32832 608D2FD8  104  Audit Process

Chunk Elements:
Address  Size  Parent   Name
62D80DA8    16 62D7BFD0 (Managed Chunk )
62D80DB8    16 62D7BFD0 (Managed Chunk )
62D80DC8    16 62D7BFD0 (Managed Chunk )
62D80DD8    16 62D7BFD0 (Managed Chunk )
62D80DE8    16 62D7BFD0 (Managed Chunk )
62E8FD60   216 62E8F888 (IPC Message He)
```

Table 104 describes the significant fields shown in the display.

Table 104 show memory debug leaks chunks Field Descriptions

Field	Description
Address	Hexadecimal address of the leaked block.
Size	Size of the leaked block (in bytes).
Alloc_pc	Address of the system call that allocated the block.
PID	The process identifier of the process that allocated the block.
Name	The name of the process that allocated the block.
Size	(Chunk Elements) Size of the leaked element (bytes).
Parent	(Chunk Elements) Parent chunk of the leaked chunk.
Name	(Chunk Elements) The name of the leaked chunk.

show memory debug leaks largest Command Example

The following example shows output from the **show memory debug leaks largest** command:

```
Router# show memory debug leaks largest

Adding blocks for GD...

                PCI memory
Alloc_pc    total leak size
```

```
                I/O memory
Alloc_pc    total leak size

                Processor memory
Alloc_pc    total leak size
608D2848    32776     inconclusive
608D2FD8    32776     inconclusive
60C656AC    288       inconclusive
60C67D74    48        inconclusive
605B7E98    40        inconclusive
605B7EDC    40        inconclusive
60C659D4    40        inconclusive
605B7E70    32        inconclusive
605B7EB4    32        inconclusive
60616750    24        inconclusive
```

The following example shows output from the second invocation of the **show memory debug leaks largest** command:

```
Router# show memory debug leaks largest

Adding blocks for GD...

                PCI memory
Alloc_pc    total leak size

                I/O memory
Alloc_pc    total leak size

                Processor memory
Alloc_pc    total leak size
608D2848    32776
608D2FD8    32776
60C656AC    288
60C67D74    48
605B7E98    40
605B7EDC    40
60C659D4    40
605B7E70    32
605B7EB4    32
60616750    24
```

show memory debug leaks summary Command Example

The following example shows output from the **show memory debug leaks summary** command:

```
Router# show memory debug leaks summary

Adding blocks for GD...

                PCI memory

Alloc PC        Size     Blocks      Bytes    What

                I/O memory

Alloc PC        Size     Blocks      Bytes    What

                Processor memory

Alloc PC        Size     Blocks      Bytes    What
```

```
0x605B7E70 0000000032 0000000001 0000000032    Init
0x605B7E98 0000000040 0000000001 0000000040    Init
0x605B7EB4 0000000032 0000000001 0000000032    Init
0x605B7EDC 0000000040 0000000001 0000000040    Init
0x60616750 0000000024 0000000001 0000000024    Init
0x606167A0 0000000024 0000000001 0000000024    Init
0x608D2848 0000032776 0000000001 0000032776    Audit Process
0x608D2FD8 0000032776 0000000001 0000032776    Audit Process
0x60C656AC 0000000040 0000000001 0000000040    Init
0x60C656AC 0000000248 0000000001 0000000248    Init
0x60C659D4 0000000040 0000000001 0000000040    Init
0x60C67D74 0000000048 0000000001 0000000048    Init
```

Table 105 describes the significant fields shown in the display.

Table 105 show memory debug leaks summary Field Descriptions

Field	Description
Alloc_pc	Address of the system call that allocated the block.
Size	Size of the leaked block.
Blocks	Number of blocks leaked.
Bytes	Total amount of memory leaked.
What	Name of the process that owns the block.

Related Commands

Command	Description
set memory debug incremental starting-time	Sets the current time as the starting time for incremental analysis.
show memory debug incremental allocation	Displays all memory blocks that were allocated after the issue of the **set memory debug incremental starting-time** command.
show memory debug incremental leaks	Displays only memory that was leaked after the issue of the **set memory debug incremental starting-time** command.
show memory debug incremental leaks lowmem	Forces incremental memory leak detection to work in low memory mode. Displays only memory that was leaked after the issue of the **set memory debug incremental starting-time** command.
show memory debug incremental status	Displays if the starting point of incremental analysis has been defined and the time elapsed since then.

show memory debug references

To display debug information on references, use the **show memory debug references** command in user EXEC or privileged EXEC mode.

show memory debug references [**dangling** [*start-address start-address*]]

Syntax Description

dangling	(Optional) Displays the possible references to free memory.
start-address	(Optional) Address numbers <0-4294967295> that determine the address range.

Command Modes

User EXEC
Privileged EXEC

Command History

Release	Modification
12.0	This command was introduced.

Examples

The following is sample output from the **show memory debug references** command:

```
Router# show memory debug references 2 3

Address  Reference Cont_block Cont_block_name
442850BC         2  44284960   bss
44285110         3  44284960   bss
4429C33C         2  44284960   bss
4429C34C         2  44284960   bss
4429C35C         3  44284960   bss
.
.
.
```

The following is sample output from the **show memory debug references dangling** command:

```
Router# show memory debug references dangling

Address  Reference Free_block Cont_block Cont_block_name
442D5774 458CE5EC  458CE5BC   44284960   bss
442D578C 46602998  46602958   44284960   bss
442D58A0 465F9BC4  465F9B94   44284960   bss
442D58B8 4656785C  4656781C   44284960   bss
442D5954 45901E7C  45901E4C   44284960   bss
.
.
.
```

Table 106 describes the significant fields shown in the displays.

Table 106 show memory debug references Field Descriptions

Field	Description
Address	Hexadecimal address of the block having the given or dangling reference.
Reference	Address which is given or dangling.
Free_block	Address of the free block which now contains the memory referenced by the dangling reference.
Cont_block	Address of the control block which contains the block having the reference.
Cont_block_name	Name of the control block.

show memory debug unused

To display debug information on leaks that are accessible, but are no longer needed, use the **show memory debug unused** command in user EXEC or privileged EXEC mode.

show memory debug unused

Syntax Description

This command has no arguments or keywords.

Command Modes

User EXEC
Privileged EXEC

Command History

Release	Modification
12.0	This command was introduced.

Examples

The following is sample output from the **show memory debug unused** command:

```
Router# show memory debug unused

Address  Alloc_pc PID  size     Name
654894B8 62BF31DC -2   44       *Init*
6549A074 601F7A84 -2   4464     XDI data
6549B218 601F7274 -2   4500     XDI data
6549DFB0 6089DDA4 42   84       Init
65509160 6089DDA4 1    84       *Init*
6550A260 6089DDA4 2    84       *Init*
6551FDB4 6089DDA4 4    84       *Init*
6551FF34 627EFA2C -2   24       *Init*
65520B3C 6078B1A4 -2   24       Parser Mode Q1
65520B88 6078B1C8 -2   24       Parser Mode Q2
65520C40 6078B1A4 -2   24       Parser Mode Q1
65520C8C 6078B1C8 -2   24       Parser Mode Q2
65520D44 6078B1A4 -2   24       Parser Mode Q1
65520D90 6078B1C8 -2   24       Parser Mode Q2
65520E48 6078B1A4 -2   24       Parser Mode Q1
65520E94 6078B1C8 -2   24       Parser Mode Q2
65520F4C 6078B1A4 -2   24       Parser Mode Q1
65520F98 6078B1C8 -2   24       Parser Mode Q2
65521050 6078B1A4 -2   24       Parser Mode Q1
6552109C 6078B1C8 -2   24       Parser Mode Q2
65521154 6078B1A4 -2   24       Parser Mode Q1
655211A0 6078B1C8 -2   24       Parser Mode Q2
.
.
.
```

Table 107 describes the significant fields shown in the display.

Table 107 show memory debug unused Field Descriptions

Field	Description
Address	Hexadecimal address of the block.
Alloc_pc	Address of the program counter that allocated the block.
PID	Process identifier of the process that allocated the block.
size	Size of the unused block (in bytes).
Name	Name of the process that owns the block.

show memory ecc

To display single-bit Error Code Correction (ECC) error logset data, use the **show memory ecc** command in privileged EXEC mode.

show memory ecc

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Command History

Release	Modification
11.1(30)CC	This command was introduced in Cisco IOS Release 11.1(30)CC.
12.0(4)XE	This command was integrated into Cisco IOS Release 12.0(4)XE.
12.0(6)S	This command was integrated into Cisco IOS Release 12.0(6)S.
12.1(13)	This command was integrated into Cisco IOS Release 12.1(13).

Usage Guidelines

Use this command to determine if the router has experienced single-bit parity errors.

Examples

The following is sample output from the **show memory ecc** command from a 12000-series router running Cisco IOS Release 12.0(23)S:

```
Router# show memory ecc
ECC Single Bit error log
------------------------
Single Bit error detected and corrected at 0x574F3640
- Occured 1 time(s)
- Whether a scrub was attempted at this address: Yes
- Syndrome of the last error at this address: 0xE9
- Error detected on a read-modify-write cycle ? No
- Address region classification: Unknown
- Address media classification : Read/Write Single Bit error detected and corrected at
0x56AB3760
- Occured 1 time(s)
- Whether a scrub was attempted at this address: Yes
- Syndrome of the last error at this address: 0x68
- Error detected on a read-modify-write cycle ? No
- Address region classification: Unknown
- Address media classification : Read/Write

Total Single Bit error(s) thus far: 2
```

Table 108 describes the significant fields shown in the first section of the display.

Table 108 show memory ecc Field Descriptions

Field	Description
Occured *n* time(s)	Number of single-bit errors that has occurred.
Whether a scrub was attempted at this address:	Indicates whether a scrub has been performed.
Syndrome of the last error at this address:	Describes the syndrome of last error.
Error detected on a read-modify-write cycle ?	Indicates whether an error has occurred.
Address region classification: Unknown	Describes the region of the error.
Address media classification :	Describes the media of the error and correction.

Related Commands

Command	Description
show memory	Displays statistics about memory, including memory-free pool statistics.

show memory failures alloc

To display statistics about failed memory allocation requests, use the **show memory failures alloc** command in the privileged EXEC mode.

show memory failures alloc

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0	This command was introduced.

Examples

The following is sample output from the **show memory failures alloc** command:

```
Router# show memory failures alloc

Caller       Pool          Size   Alignment   When
0x60394744  I/O             1684     32      00:10:03
0x60394744  I/O             1684     32      00:10:03
0x60394744  I/O             1684     32      00:10:03
0x60394744  I/O             1684     32      00:10:03
0x60394744  I/O             1684     32      00:10:03
0x60394744  I/O             1684     32      00:10:03
0x60394744  I/O             1684     32      00:10:03
0x60394744  I/O             1684     32      00:10:03
0x60394744  I/O             1684     32      00:10:04
0x60394744  I/O             1684     32      00:10:04
```

Table 109 describes the significant fields shown in the display.

Table 109 show memory failures alloc Field Descriptions

Field	Description
Caller	Address of the allocator function that issued memory allocation request that failed.
Pool	Pool from which the memory was requested.
Size	Size of the memory requested in bits.
Alignment	Memory alignment in bits.
When	Time of day at which the memory allocation request was issued.

show memory fast

To display fast memory details for the router, use the **show memory fast** command.

show memory fast [**allocating-process** [**totals**] | **dead** [**totals**] | **free** [**totals**]]

Syntax Description

allocating-process	(Optional) Include allocating process names with the standard output.
dead	(Optional) Display only memory owned by dead processes.
free	(Optional) Display only memory not allocated to a process.
totals	(Optional) Summarizes the statistics for allocating processes, dead memory, or free memory.

Command Modes

Exec

Command History

Release	Modification
12.1	This command was introduced in a release prior to 12.1. This command replaced the **show memory sram** command.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

The show memory fast command displays the statistics for the fast memory. “Fast memory” is another name for “processor memory,” and is also known as “cache memory.” Cache memory is called fast memory because the processor can generally access the local cache (traditionally stored on SRAM positioned close to the processor) much more quickly than main memory or RAM.

Note The **show memory fast** command is a command alias for the **show memory processor** command. These commands will issue the same output.

Examples

The following example shows sample output from the **show memory fast** and the **show memory processor** commands:

```
Router>show memory fast

          Processor memory

 Address      Bytes     Prev     Next Ref     PrevF    NextF Alloc PC  what
8404A580 0001493284 00000000 841B6ECC 000  0        84BADF88 815219D8  (coalesced)
841B6ECC 0000020004 8404A580 841BBD18 001  -------- -------- 815DB094  Managed Chunk Queue
Elements
841BBD18 0000001504 841B6ECC 841BC320 001  -------- -------- 8159EAC4  List Elements
841BC320 0000005004 841BBD18 841BD6D4 001  -------- -------- 8159EB04  List Headers
841BD6D4 0000000048 841BC320 841BD72C 001  -------- -------- 81F2A614  *Init*
841BD72C 0000001504 841BD6D4 841BDD34 001  -------- -------- 815A9514  messages
841BDD34 0000001504 841BD72C 841BE33C 001  -------- -------- 815A9540  Watched messages
841BE33C 0000001504 841BDD34 841BE944 001  -------- -------- 815A95E4  Watched Semaphore
```

```
841BE944 0000000504 841BE33C 841BEB64 001  -------- -------- 815A9630  Watched Message
Queue
841BEB64 0000001504 841BE944 841BF16C 001  -------- -------- 815A9658  Watcher Message
Queue
841BF16C 0000001036 841BEB64 841BF5A0 001  -------- -------- 815A2B24  Process Array
-- More --
<Ctrl+z>

Router>show memory processor

          Processor memory

 Address      Bytes     Prev     Next Ref     PrevF    NextF Alloc PC  what
8404A580 0001493284 00000000 841B6ECC 000  0        84BADF88 815219D8  (coalesced)
841B6ECC 0000020004 8404A580 841BBD18 001  -------- -------- 815DB094  Managed Chunk Queue
Elements
841BBD18 0000001504 841B6ECC 841BC320 001  -------- -------- 8159EAC4  List Elements
841BC320 0000005004 841BBD18 841BD6D4 001  -------- -------- 8159EB04  List Headers
841BD6D4 0000000048 841BC320 841BD72C 001  -------- -------- 81F2A614  *Init*
841BD72C 0000001504 841BD6D4 841BDD34 001  -------- -------- 815A9514  messages
841BDD34 0000001504 841BD72C 841BE33C 001  -------- -------- 815A9540  Watched messages
841BE33C 0000001504 841BDD34 841BE944 001  -------- -------- 815A95E4  Watched Semaphore
841BE944 0000000504 841BE33C 841BEB64 001  -------- -------- 815A9630  Watched Message
Queue
841BEB64 0000001504 841BE944 841BF16C 001  -------- -------- 815A9658  Watcher Message
Queue
841BF16C 0000001036 841BEB64 841BF5A0 001  -------- -------- 815A2B24  Process Array
-- More --
<Ctrl+z>

Router>
```

The following example shows sample output from the **show memory fast allocating-process** command, followed by sample output from the **show memory fast allocating-process totals** command:

```
Router#show memory fast allocating-process

          Processor memory

 Address      Bytes     Prev     Next Ref   Alloc Proc     Alloc PC  What
8404A580 0001493284 00000000 841B6ECC 000                  815219D8  (coalesced)
841B6ECC 0000020004 8404A580 841BBD18 001  *Init*          815DB094  Managed Chunk Queue
Elements
841BBD18 0000001504 841B6ECC 841BC320 001  *Init*          8159EAC4  List Elements
841BC320 0000005004 841BBD18 841BD6D4 001  *Init*          8159EB04  List Headers
841BD6D4 0000000048 841BC320 841BD72C 001  *Init*          81F2A614  *Init*
841BD72C 0000001504 841BD6D4 841BDD34 001  *Init*          815A9514  messages
841BDD34 0000001504 841BD72C 841BE33C 001  *Init*          815A9540  Watched messages
841BE33C 0000001504 841BDD34 841BE944 001  *Init*          815A95E4  Watched Semaphore
841BE944 0000000504 841BE33C 841BEB64 001  *Init*          815A9630  Watched Message Queue
841BEB64 0000001504 841BE944 841BF16C 001  *Init*          815A9658  Watcher Message Queue
841BF16C 0000001036 841BEB64 841BF5A0 001  *Init*          815A2B24  Process Array
 --More--
<Ctrl+z>

c2600-1#show memory fast allocating-process totals

Allocator PC Summary for: Processor

    PC          Total   Count  Name
0x815C085C    1194600     150  Process Stack
0x815B6C28     948680       5  pak subblock chunk
```

```
0x819F1DE4     524640       8  BGP (0) update
0x815C4FD4     393480       6  MallocLite
0x815B5FDC     351528      30  TW Buckets
0x819F14DC     327900       5  connected
0x81A1E838     327900       5  IPv4 Unicast net-chunk(8)
0x8153DFB8     248136     294  *Packet Header*
0x82142438     133192       4  CEF: 16 path chunk pool
0x82151E0C     131116       1  Init
0x819F1C8C     118480       4  BGP (0) attr
0x815A4858     100048     148  Process
0x8083DA44      97248      17


 --More--
<Ctrl+z>
```

The following example shows sample output from the **show memory fast dead** command:

```
Router#show memory fast dead

          Processor memory

 Address      Bytes     Prev     Next Ref     PrevF    NextF Alloc PC  what
8498FC20 0000000028 8498FB90 8498FC64 001  -------- -------- 81472B24  AAA MI SG NAME
        -------
             68
Router#show memory fast dead totals

Dead Proc Summary for: Processor

    PC          Total   Count  Name
0x81472B24         68       1  AAA MI SG NAME

Router#
```

show memory fragment

To display the block details of fragmented free blocks and allocated blocks, which is physically just before or after the blocks on the free list, use the **show memory fragment** command in user EXEC or privileged EXEC mode.

show memory [**processor** | **io**] **fragment** [**detail**]

Syntax Description

processor	(Optional) Displays the processor memory information.
io	(Optional) Displays the I/O memory information.
fragment	Displays the information of the free blocks and the blocks surrounding the free blocks.
detail	(Optional) Displays the detailed information of all the free blocks and the blocks surrounding the free blocks that are located between the allocated blocks.

Command Modes

User EXEC
Privileged EXEC

Command History

Release	Modification
12.3(14)T	This command was introduced.
12.2(33)SRB	This command was integrated into Cisco IOS Release 12.2(33)SRB.

Examples

The following is sample output from the **show memory processor fragment** command:

```
Router# show memory processor fragment

Processor memory
Free memory size : 65516944 Number of free blocks:      230
Allocator PC Summary for allocated blocks in pool: Processor

    PC          Total   Count  Name
0x6047DDCC     852020       1  atmdx_vc_table
0x6075DC30     544392       4  ATM1/0
0x61BDBA14     131176       2  eddri_self_event
0x61913BEC     131124       1  l2tp tnl table
0x602E9820     114832       1  AutoVC Msg Chunk
0x6071253C      98408       2  Exec
0x607DF5BC      96624      12  Process Stack
0x6118DDA0      77252       1  Spanning Tree Opt Port Block
0x61F13C30      67636       1  QOS_MODULE_MAIN
0x6047DD3C      65640       2  atmdx_tx_shadow
0x614B6624      65588       1  CEF: loadinfo chunk
0x614D1924      65588       1  IP mtrie node
0x614A58A0      65588       1  CEF: 16 path chunk pool
0x619241D4      65588       1  PPTP mgd timer chunk
0x606581CC      65588       1  AAA DB Chunk
0x607E5EAC      65588       1  MallocLite
0x6192420C      65588       1  PPTP: pptp_tunneltype chunk
0x6075DCB8      45924      10  FastEthernet2/
```

```
0x607CA400      36288       2  pak subblock chunk
0x6255648C      28948       1  CCPROXY_CT
0x6047DD7C      24628       1  atmdx_bfd_cache
0x6047DAA4      23500       1  atmdx_instance
0x6047DAE8      23500       1  atmdx_instance snap
0x60962DFC      21420      17  TCP CB
0x616F729C      20052       1  AC context chunks
0x616F72C8      20052       1  AC Mgr mgd timer chunk
0x60734010      16644      19  *Packet Header*
0x6047DE0C      16436       1  atmdx_abr_stats
0x6047DCFC      16112       2  atmdx_rx_pool_info
0x60A77E98      13060       1  DHCPD Message Workspace
0x61F50008      12852       1  CCVPM_HTSP
0x60D509BC      12580      17  Virtual Exec
0x60EFA1EC      12344       1  RSVP DB Handle Bin
.
.
.
0x6067AE44         76       1  AAA Secrettype encrypt
0x61C0EEC0         76       1  Init
0x60F76B1C         76       1  SNMP Trap
0x60BE2444         76       1  Init
0x62638F78         76       1  EEM ED Syslog
0x6077C574         76       1  Init
0x608F7030         76       1  IPC Name String
0x608EEAB8         76       1  IPC Name
0x620468A8         76       1  ivr: ccapAppEntry_t name
0x6066D084         76       1  gk process
0x6064824C         76       1  AAA MI SG NAME

Allocator PC Summary for free blocks in pool: Processor

    PC          Total   Count  Name
0x6071253C   67387912       2  (fragment)
0x60734010   63292440      11  *Packet Header*
0x60962DFC     105552      10  (coalesced)
0x60D509BC      98384      10  (coalesced)
0x60D4A0B4      70776       9  (coalesced)
0x60803260      21488       4  (fragment)
0x60B2E488      19704       2  (fragment)
0x606E0278      19272       1  (coalesced)
0x606DD8D8       9024     113  Init
0x60B27FE8       5740       3  (fragment)
0x60778AAC       3504       1  (coalesced)
0x607AC764       2212      11  Process Events
0x60F7FCD4       1556       9  (fragment)
0x6071F3FC       1316      12  (fragment)
0x606C5324       1176       6  (coalesced)
0x60D7C518       1148       1  (coalesced)
0x624E170C        876       1  (coalesced)
0x60A68164        588       3  (fragment)
0x60B302C0        408       5  (fragment)
0x60976574        272       2  AAA Event Data
0x60801E38        216       2  (fragment)
0x611DA23C        164       1  shelf_info
0x60A6A638        148       1  (fragment)
0x60801D2C        148       1  (fragment)
0x60D29DCC        148       1  (fragment)
0x62628CA0        144       1  (fragment)
0x60A68218        104       1  (fragment)
0x606B9614         88       1  NameDB String
0x6090A978         84       1  (fragment)
0x606C51D0         84       1  (fragment)
0x62647558         76       1  (fragment)
```

The following is sample output from the **show memory processor fragment detail** command:

```
Router# show memory processor fragment detail

Processor memory
Free memory size : 65566148 Number of free blocks:      230
 Address      Bytes     Prev     Next Ref     PrevF    NextF Alloc PC  what
645A8148 0000000028 645A80F0 645A8194 001  -------- -------- 60695B20  Init
645A8194 0000000040 645A8148 645A81EC 000  0        200B4300 606B9614  NameDB String
645A81EC 0000000260 645A8194 645A8320 001  -------- -------- 607C2D20  Init
200B42B4 0000000028 200B4268 200B4300 001  -------- -------- 62366C80  Init
200B4300 0000000028 200B42B4 200B434C 000  645A8194 6490F7E8 60976574  AAA Event Data
200B434C 0000002004 200B4300 200B4B50 001  -------- -------- 6267D294  Coproc Request
Structures
6490F79C 0000000028 6490F748 6490F7E8 001  -------- -------- 606DDA04  Parser Linkage
6490F7E8 0000000028 6490F79C 6490F834 000  200B4300 6491120C 606DD8D8  Init
6490F834 0000006004 6490F7E8 64910FD8 001  -------- -------- 607DF5BC  Process Stack
649111A0 0000000060 64911154 6491120C 001  -------- -------- 606DE82C  Parser Mode
6491120C 0000000028 649111A0 64911258 000  6490F7E8 500770F0 606DD8D8  Init
64911258 0000000200 6491120C 64911350 001  -------- -------- 603F0E38  Init
.
.
.
504DCF54 0000001212 504DB2E4 504DD440 001  -------- -------- 60962DFC  TCP CB
2C41DCA4 0000000692 2C41BCC8 2C41DF88 001  -------- -------- 60D509BC  Virtual Exec
2C41DF88 0000005344 2C41DCA4 2C41F498 000  504DB2E4 6449A828 60D509BC  (coalesced)
2C41F498 0000000692 2C41DF88 2C41F77C 001  -------- -------- 60D509BC  Virtual Exec
6449A544 0000000692 64499794 6449A828 001  -------- -------- 60D509BC  Virtual Exec
6449A828 0000007760 6449A544 6449C6A8 000  2C41DF88 504D89D4 60D509BC  (coalesced)
6449C6A8 0000008044 6449A828 6449E644 001  -------- -------- 60D2AACC  Virtual Exec
504D8778 0000000556 504D754C 504D89D4 001  -------- -------- 60D4A0B4  Virtual Exec
504D89D4 0000009860 504D8778 504DB088 000  6449A828 504D1B78 60D4A0B4  (coalesced)
504DB088 0000000556 504D89D4 504DB2E4 001  -------- -------- 60D4A0B4  Virtual Exec
504D168C 0000001212 504C9658 504D1B78 001  -------- -------- 60962DFC  TCP CB
504D1B78 0000008328 504D168C 504D3C30 000  504D89D4 504C5B54 60962DFC  (coalesced)
504D3C30 0000001212 504D1B78 504D411C 001  -------- -------- 60962DFC  TCP CB
504C5870 0000000692 504C5504 504C5B54 001  -------- -------- 60D509BC  Virtual Exec
504C5B54 0000005344 504C5870 504C7064 000  504D1B78 2C423A88 60D509BC  (coalesced)
504C7064 0000000408 504C5B54 504C722C 001  -------- -------- 606E0E44  Chain Cache No
2C42359C 0000001212 2C41F77C 2C423A88 001  -------- -------- 60962DFC  TCP CB
2C423A88 0000008328 2C42359C 2C425B40 000  504C5B54 504D411C 60962DFC  (coalesced)
504E7DD8 0000000828 504E2660 504E8144 001  -------- -------- 60734010  *Packet Header*
65006A08 0000000408 65003834 65006BD0 001  -------- -------- 606E0E44  Chain Cache No
65006BD0 0000020520 65006A08 6500BC28 000  504E2660 0        60803260  (coalesced)
6500BC28 0000000828 65006BD0 6500BF94 001  -------- -------- 60734010  *Packet Header*
5C3AE7B8 0000000828 5C3AE614 5C3AEB24 001  -------- -------- 60734010  *Packet Header*
5C3AEB24 0063247532 5C3AE7B8 20000000 000  0        6500C300 60734010  (coalesced)
20000000 0000000828 5C3AEB24 2000036C 001  -------- -------- 60734010  *Packet Header*
6500BF94 0000000828 6500BC28 6500C300 001  -------- -------- 60734010  *Packet Header*
6500C300 0004760912 6500BF94 50000000 000  5C3AEB24 2C42E310 6071253C  (coalesced)
50000000 0000000828 6500C300 5000036C 001  -------- -------- 60734010  *Packet Header*
2C42E0B4 0000000556 2C429430 2C42E310 001  -------- -------- 60D4A0B4  Virtual Exec
2C42E310 0062725312 2C42E0B4 00000000 000  6500C300 0        6071253C  (coalesced
```

Related Commands

Command	Description
memory io	Configures thresholds for I/O memory.
memory processor	Configures thresholds for processor memory.

show memory multibus

To display statistics about multibus memory, including memory-free pool statistics, use the **show memory multibus** command in user EXEC or privileged EXEC mode.

show memory multibus [allocating-process [totals]| dead [totals]| free [totals]]

Syntax Description

allocating-process [totals]	(Optional) Displays allocating memory totals by name.
dead [totals]	(Optional) Displays memory totals on dead processes.
fragment **[detail]**	(Optional) Displays memory statistics for fragmented processes.
free [totals]	(Optional) Displays statistics on free memory.
statistics **[history]**	(Optional) Displays memory pool history statistics on all processes.

Command Modes

User EXEC
Privileged EXEC

Command History

Release	Modification
12.0	This command was introduced.

Examples

The following is sample output from the **show memory multibus** command:

```
Router# show memory multibus

          Processor memory

 Address      Bytes     Prev     Next Ref     PrevF    NextF Alloc PC  what
6540BBA0 0000016388 00000000 6540FBD4 001  -------- -------- 60883984  TW Buckes
6540FBD4 0000016388 6540BBA0 65413C08 001  -------- -------- 60883984  TW Buckes
65413C08 0000016388 6540FBD4 65417C3C 001  -------- -------- 60883984  TW Buckes
65417C3C 0000006004 65413C08 654193E0 001  -------- -------- 608A0D4C  Process k
654193E0 0000012004 65417C3C 6541C2F4 001  -------- -------- 608A0D4C  Process k
6541C2F4 0000411712 654193E0 65480B64 000  0        0        608A0D4C  (fragmen)
65480B64 0000020004 6541C2F4 654859B8 001  -------- -------- 608CF99C  Managed s
654859B8 0000010004 65480B64 654880FC 001  -------- -------- 6085C7F8  List Eles
654880FC 0000005004 654859B8 654894B8 001  -------- -------- 6085C83C  List Heas
654894B8 0000000048 654880FC 65489518 001  -------- -------- 62BF31DC  *Init*
.
.
.
```

Table 110 describes the significant fields shown in the display.

Table 110 show memory multibus Field Descriptions

Field	Description
Address	Hexadecimal address of the block.
Bytes	Size of the block (in bytes).

Table 110 show memory multibus Field Descriptions (continued)

Field	Description
Prev	Address of the preceding block (should match the address on the preceding line).
Next	Address of the following block (should match the address on the following line).
Ref	Reference count for that memory block, indicating how many different processes are using that block of memory.
PrevF	Address of the preceding free block (if free).
NextF	Address of the following free block (if free).
Alloc PC	Address of the program counter that allocated the block.
What	Name of the process that owns the block, or "(fragmen)" if the block is a fragment, or "(coalesced)" if the block was coalesced from adjacent free blocks.

show memory pci

To display statistics about Peripheral Component Interconnect (PCI) memory, use the **show memory pci** command in user EXEC or privileged EXEC mode.

show memory pci

Syntax Description

This command has no arguments or keywords.

Command Modes

User EXEC
Privileged EXEC

Command History

Release	Modification
12.0	This command was introduced.

Examples

The following is sample output from the **show memory pci** command:

```
Router# show memory pci

          I/O memory

 Address      Bytes     Prev     Next Ref     PrevF    NextF Alloc PC  what
0E000000 0000000032 00000000 0E000050 000  64F5EBF4 0        00000000  (fragmen)
0E000050 0000000272 0E000000 0E000190 001  -------- -------- 607E2EC0  *Packet *
0E000190 0000000272 0E000050 0E0002D0 001  -------- -------- 607E2EC0  *Packet *
0E0002D0 0000000272 0E000190 0E000410 001  -------- -------- 607E2EC0  *Packet *
0E000410 0000000272 0E0002D0 0E000550 001  -------- -------- 607E2EC0  *Packet *
0E000550 0000000272 0E000410 0E000690 001  -------- -------- 607E2EC0  *Packet *
0E000690 0000000272 0E000550 0E0007D0 001  -------- -------- 607E2EC0  *Packet *
0E0007D0 0000000272 0E000690 0E000910 001  -------- -------- 607E2EC0  *Packet *
0E000910 0000000272 0E0007D0 0E000A50 001  -------- -------- 607E2EC0  *Packet *
0E000A50 0000000272 0E000910 0E000B90 001  -------- -------- 607E2EC0  *Packet *
0E000B90 0000000272 0E000A50 0E000CD0 001  -------- -------- 607E2EC0  *Packet *
 Address      Bytes     Prev     Next Ref     PrevF    NextF Alloc PC  what
0E000CD0 0000000272 0E000B90 0E000E10 001  -------- -------- 607E2EC0  *Packet *
0E000E10 0000000272 0E000CD0 0E000F50 001  -------- -------- 607E2EC0  *Packet *
```

Table 111 describes the significant fields shown in the display.

Table 111 show memory pci Field Descriptions

Field	Description
Address	Hexadecimal address of the block.
Bytes	Size of the block (in bytes).
Prev	Address of the preceding block (should match the address on the preceding line).
Next	Address of the following block (should match the address on the following line).
Ref	Reference count for that memory block, indicating how many different processes are using that block of memory.

Table 111 show memory pci Field Descriptions (continued)

Field	Description
PrevF	Address of the preceding free block (if free).
NextF	Address of the following free block (if free).
Alloc PC	Address of the program counter that allocated the block.
what	Name of process that owns the block, or “(fragmen)” if the block is a fragment, or "(coalesced)" if the block was coalesced from adjacent free blocks.

show memory processor

To display statistics on the router processor memory, use the **show memory processor** command in user EXEC or privileged EXEC mode.

show memory processor [**fragment** | **free** | **statistics**]

Syntax Description

fragment	(Optional) Displays the block details of fragmented free blocks and allocated blocks, which are shown either preceding or following the blocks on the free list.
free	(Optional) Displays the number of free blocks.
statistics	(Optional) Displays only memory processor statistics.

Command Modes

User EXEC
Privileged EXEC

Command History

Release	Modification
12.0	This command was introduced.

Examples

The following is sample output from the **show memory processor** commands:

```
Router# show memory processor

          Processor memory

 Address      Bytes     Prev     Next Ref     PrevF    NextF Alloc PC  what
6540BBA0 0000016388 00000000 6540FBD4 001  -------- -------- 60883984  TW Buckes
6540FBD4 0000016388 6540BBA0 65413C08 001  -------- -------- 60883984  TW Buckes
65413C08 0000016388 6540FBD4 65417C3C 001  -------- -------- 60883984  TW Buckes
65417C3C 0000006004 65413C08 654193E0 001  -------- -------- 608A0D4C  Process k
654193E0 0000012004 65417C3C 6541C2F4 001  -------- -------- 608A0D4C  Process k
6541C2F4 0000411712 654193E0 65480B64 000  0        0        608A0D4C  (fragmen)
65480B64 0000020004 6541C2F4 654859B8 001  -------- -------- 608CF99C  Managed s
654859B8 0000010004 65480B64 654880FC 001  -------- -------- 6085C7F8  List Eles
654880FC 0000005004 654859B8 654894B8 001  -------- -------- 6085C83C  List Heas
654894B8 0000000048 654880FC 65489518 001  -------- -------- 62BF31DC  *Init*
```

Table 112 describes the significant fields shown in the display.

Table 112 show memory processor Field Descriptions

Field	Description
Address	Hexadecimal address of the block.
Bytes	Size of the block (in bytes).
Prev.	Address of the preceding block (should match the address on the preceding line).
Next	Address of the following block (should match the address on the following line).
Ref	Reference count for that memory block, indicating how many different processes are using that block of memory.

Table 112 show memory processor Field Descriptions (continued)

Field	Description
PrevF	Address of the preceding free block (if free).
NextF	Address of the following free block (if free).
Alloc PC	Address of the program counter that allocated the block.
What	Name of the process that owns the block, or "(fragmen)" if the block is a fragment, or "(coalesced)" if the block was coalesced from adjacent free blocks.

The following is sample output from the **show memory processor fragment** command:

```
Router# show memory processor fragment

          Processor memory

Free memory size :  3144348 Number of free blocks:       96

Allocator PC Summary for allocated blocks in pool: Processor

    PC          Total   Count  Name
0x6069A038     262196       1  TACL FLT
0x62224AA8     219188       1  QOS_MODULE_MAIN
0x61648840     131124       1  Init
0x6218DAA4      73780       1  CCSIP_UDP_SOCKET
0x61649288      65588       1  CEF: loadinfo chunk
0x61BFD4B8      65588       1  PPTP mgd timer chunk
0x61EE1050      65588       1  eddri_self_event
0x607C13C4      49204       1  Exec
0x608A0D4C      35208       4  Process Stack
0x6069D804      32052       1  TACL hist
0x61631A90      21444       2  CEF: IPv4 Unicast RPF subblock
0x62BA5DD8      20432       1  Init
0x6086F858      20052       1  RMI-RO_RU Chun
0x608CF99C      20052       1  Managed Chunk Queue Elements
```

Table 113 describes the significant fields shown in the display.

Table 113 show memory processor fragment Field Descriptions

Field	Description
PC	Program counter
Total	Total memory allocated by the process (in bytes).
Count	Number of allocations.
Name	Name of the allocating process.

The following is sample output from the **show memory processor free** command:

```
Router# show memory processor free

          Processor memory

 Address      Bytes     Prev     Next Ref     PrevF    NextF Alloc PC  what

             24    Free list 1
66994680 0000000072 66994618 669946FC 000  0        6698FFC8 60699114  Turbo ACr
6698FFC8 0000000072 6698FF60 66990044 000  66994680 659CF6B0 60699114  Turbo ACr
```

```
659CF6B0 0000000024 659CF678 659CF6FC 000  6698FFC8 659CF86C 6078A2CC  Init
659CF86C 0000000024 659CF710 659CF8B8 000  659CF6B0 65ADB53C 6078A2CC  Init
65ADB53C 0000000024 65ADB504 65ADB588 000  659CF86C 65ADFC38 6078A2CC  Init
65ADFC38 0000000024 65ADFC00 65ADFC84 000  65ADB53C 65B6C504 6078A2CC  Init
65B6C504 0000000024 65B6C4B8 65B6C550 000  65ADFC38 6593E924 6078A2CC  Init
6593E924 0000000028 6593E8E8 6593E974 000  65B6C504 65CCB054 6078A2CC  Init
65CCB054 0000000024 65CCB01C 65CCB0A0 000  6593E924 65CCBD98 6078A2CC  Init
65CCBD98 0000000028 65CCBD60 65CCBDE8 000  65CCB054 65CCFB70 6078A2CC  Init
65CCFB70 0000000024 65CCFB38 65CCFBBC 000  65CCBD98 65D0BB58 6078A2CC  Init
65D0BB58 0000000024 65D0BB20 65D0BBA4 000  65CCFB70 65D0C5F0 6078A2CC  Init
65D0C5F0 0000000024 65D0C5B8 65D0C63C 000  65D0BB58 65CFF2F4 6078A2CC  Init
65CFF2F4 0000000024 65CFF2BC 65CFF340 000  65D0C5F0 6609B7B8 6078A2CC  Init
6609B7B8 0000000036 6609AFC8 6609B810 000  65CFF2F4 660A0BD4 6078A2CC  Init
```

Table 114 describes the significant fields shown in the display.

Table 114 show memory processor free Field Descriptions

Field	Description
Address	Hexadecimal address of the block.
Bytes	Size of the block (in bytes).
Prev	Address of the preceding block (should match the address on preceding row).
Next	Address of the following block (should match the address on following row).
Ref	Reference count for that memory block, indicating how many different processes are using that block of memory.
PrevF	Address of the preceding free block (if free).
NextF	Address of the following free block (if free).
Alloc PC	Address of the program counter that allocated the block.
what	Name of the process that owns the block, or "(fragment)" if the block is a fragment, or "(coalesced)" if the block was coalesced from adjacent free blocks.

The following is sample output from the **show memory processor statistics** command:

```
Router# show memory processor statistics

               Head    Total(b)     Used(b)     Free(b)   Lowest(b)  Largest(b)
Processor   6540BBA0   415187836    27216968   387970868   385755044   381633404
      I/O    E000000    33554432     6226336    27328096    27328096    27317852
.
.
.
```

Table 115 describes the significant fields shown in the display.

Table 115 show memory processor statistics Field Descriptions

Field	Description
Head	Hexadecimal address of the head of the memory allocation chain.
Total(b)	Sum of the used bytes plus free bytes.
Used(b)	Amount of memory in use (in bytes).
Free(b)	Amount of memory not in use (in bytes).

Table 115 show memory processor statistics Field Descriptions (continued)

Field	Description
Lowest(b)	Smallest amount of free memory since last boot (in bytes).
Largest(b)	Size of the largest available free block (in bytes).

show memory scan

To monitor the number and type of parity (memory) errors on your system, use the **show memory scan** command in EXEC mode.

show memory scan

Syntax Description

This command has no arguments or keywords.

Command Modes

EXEC

Command History

Release	Modification
12.0(4)XE	This command was introduced.
12.0(7)T	This command was implemented in Cisco IOS Release 12.0(7) T.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Examples

The following example shows a result with no memory errors:

```
Router# show memory scan

Memory scan is on.
No parity error has been detected.
```

If errors are detected in the system, the **show memory scan** command generates an error report. In the following example, memory scan detected a parity error:

```
Router# show memory scan

Memory scan is on.
Total Parity Errors 1.
Address   BlockPtr   BlckSize  Disposit  Region Timestamp
6115ABCD  60D5D090   9517A4    Scrubed   Local 16:57:09 UTC Thu  Mar 18
```

Table 116 describes the fields contained in the error report.

Table 116 show memory scan Field Descriptions

Field	Description
Address	The byte address where the error occurred.
BlockPtr	The pointer to the block that contains the error.
BlckSize	The size of the memory block

Table 116 show memory scan Field Descriptions (continued)

Field	Description
Disposit	The action taken in response to the error: • BlockInUse—An error was detected in a busy block. • InFieldPrev—An error was detected in the previous field of a block header. • InHeader—An error was detected in a block header. • Linked—A block was linked to a bad list. • MScrubed—The same address was "scrubbed" more than once, and the block was linked to a bad list. • MultiError—Multiple errors have been found in one block. • NoBlkHdr—No block header was found. • NotYet—An error was found; no action has been taken at this time. • Scrubed—An error was "scrubbed." • SplitLinked—A block was split, and only a small portion was linked to a bad list.
Region	The memory region in which the error was found: • IBSS—image BSS • IData—imagedata • IText—imagetext • local—heap
Timestamp	The time the error occurred.

show memory statistics history table

To display the history of memory consumption, use the **show memory statistics history table** command in user EXEC or privileged EXEC mode.

show memory statistics history table

Syntax Description

This command has no arguments or keywords.

Command Modes

User EXEC
Privileged EXEC

Command History

Release	Modification
12.3(14)T	This command was introduced.
12.2(33)SRB	This command was integrated into Cisco IOS Release 12.2(33)SRB.

Examples

The following is sample output from the **show memory statistics history table** command:

```
Router# show memory statistics history table

History for Processor memory

Time: 15:48:56.806
Used(b): 422748036 Largest(b): 381064952 Free blocks :291
Maximum memory users for this period
Process Name           Holding   Num Alloc
Virtual Exec             26992          37
TCP Protocols            14460           6
IP Input                  1212           1

Time: 14:42:54.506
Used(b): 422705876 Largest(b): 381064952 Free blocks :296
Maximum memory users for this period
Process Name           Holding   Num Alloc
Exec                  400012740          24
Dead                   1753456          90
Pool Manager            212796         257

Time: 13:37:26.918
Used(b): 20700520 Largest(b): 381064952 Free blocks :196
Maximum memory users for this period
Process Name           Holding   Num Alloc
Exec                      8372           5

Time: 12:39:44.422
Used(b): 20701436 Largest(b): 381064952 Free blocks :193

Time: 11:46:25.135
Used(b): 20701436 Largest(b): 381064952 Free blocks :193
Maximum memory users for this period
Process Name           Holding   Num Alloc
CDP Protocol              3752          25
```

```
Time: 10:44:24.342
Used(b): 20701400 Largest(b): 381064952 Free blocks :194

Time: 09:38:53.038
Used(b): 20701400 Largest(b): 381064952 Free blocks :194

Time: 08:33:35.154
Used(b): 20701400 Largest(b): 381064952 Free blocks :194

Time: 07:28:05.987
Used(b): 20701400 Largest(b): 381064952 Free blocks :194

Time: 06:35:22.878
Used(b): 20701400 Largest(b): 381064952 Free blocks :194

Time: 05:42:14.286
Used(b): 20701400 Largest(b): 381064952 Free blocks :194

Time: 04:41:53.486
Used(b): 20701400 Largest(b): 381064952 Free blocks :194

Time: 03:48:47.891
Used(b): 20701400 Largest(b): 381064952 Free blocks :194

Time: 02:46:32.391
Used(b): 20701400 Largest(b): 381064952 Free blocks :194

Time: 01:54:27.931
Used(b): 20717804 Largest(b): 381064952 Free blocks :189

Time: 01:02:05.535
Used(b): 20717804 Largest(b): 381064952 Free blocks :189
Maximum memory users for this period
Process Name           Holding   Num Alloc
Entity MIB API           67784          16
TTY Background           12928           4
Exec                      7704           3

Time: 00:00:17.936
Used(b): 21011192 Largest(b): 381064952 Free blocks :186
Maximum memory users for this period
Process Name           Holding   Num Alloc
Init                  18653520        6600
CCPROXY_CT              599068          57
Proxy Session Applic    275424          21

History for I/O memory

Time: 15:48:56.809
Used(b):  7455520 Largest(b): 59370080 Free blocks :164

Time: 14:42:54.508
Used(b):  7458064 Largest(b): 59370080 Free blocks :165
Maximum memory users for this period
Process Name           Holding   Num Alloc
Pool Manager            141584         257

Time: 13:37:26.920
Used(b):  7297744 Largest(b): 59797664 Free blocks :25

Time: 12:39:44.424
Used(b):  7297744 Largest(b): 59797664 Free blocks :25
```

```
Time: 11:46:25.137
Used(b):  7297744 Largest(b): 59797664 Free blocks :25

Time: 10:44:24.344
Used(b):  7297744 Largest(b): 59797664 Free blocks :25

Time: 09:38:53.040
Used(b):  7297744 Largest(b): 59797664 Free blocks :25

Time: 08:33:35.156
Used(b):  7297744 Largest(b): 59797664 Free blocks :25

Time: 07:28:05.985
Used(b):  7297744 Largest(b): 59797664 Free blocks :25

Time: 06:35:22.877
Used(b):  7297744 Largest(b): 59797664 Free blocks :25

Time: 05:42:14.285
Used(b):  7297744 Largest(b): 59797664 Free blocks :25

Time: 04:41:53.485
Used(b):  7297744 Largest(b): 59797664 Free blocks :25

Time: 03:48:47.889
Used(b):  7297744 Largest(b): 59797664 Free blocks :25

Time: 02:46:32.389
Used(b):  7297744 Largest(b): 59797664 Free blocks :25

Time: 01:54:27.929
Used(b):  7308336 Largest(b): 59797664 Free blocks :23

Time: 01:02:05.533
Used(b):  7308336 Largest(b): 59797664 Free blocks :23

Time: 00:00:17.937
Used(b):  7308336 Largest(b): 59797664 Free blocks :23
Maximum memory users for this period
Process Name           Holding   Num Alloc
Init                   7296000         214
Pool Manager               816           3
```

Related Commands

Command	Description
memory statistics history table	Changes the memory log time.

show memory transient

To display statistics about transient memory, use the **show memory transient** command in user EXEC or privileged EXEC mode.

show memory transient [**allocating-process [totals] | dead [totals] | fragment [detail] | free [totals] | statistics [history]**]

Syntax Description

allocating-process	(Optional) Displays allocating memory totals by name.
dead [totals]	(Optional) Displays memory totals on dead processes.
fragment [detail]	(Optional) Displays memory statistics for fragmented processes.
free [totals]	(Optional) Displays statistics on free memory.
statistics [history]	(Optional) Displays memory pool history statistics on all processes.

Command Modes

User EXEC
Privileged EXEC

Command History

Release	Modification
12.0	This command was introduced.

Examples

The following is sample output from the **show memory transient** command:

```
Router# show memory transient

          Processor memory

 Address      Bytes     Prev     Next Ref     PrevF    NextF Alloc PC  what
81F99C00 0002236408 00000000 821BBC28 000  829C8104 82776FD0 8060B6D0  (coalesc)
821BBC28 0000020004 81F99C00 821C0A7C 001  -------- -------- 8002D5C0  Managed s
821C0A7C 0000010004 821BBC28 821C31C0 001  -------- -------- 811604C0  List Eles
821C31C0 0000005004 821C0A7C 821C457C 001  -------- -------- 81160500  List Heas
```

Table 117 describes the significant fields shown in the display.

Table 117 show memory transient Field Descriptions

Field	Description
Address	Hexadecimal address of the block.
Bytes	Size of the block (in bytes).
Prev	Address of the preceding block (should match the address on preceding line).
Next	Address of the following block (should match the address on following line).
Ref	Reference count for that memory block, indicating how many different processes are using that block of memory.
PrevF	Address of the preceding free block (if free).

Table 117 show memory transient Field Descriptions (continued)

Field	Description
NextF	Address of the following free block (if free).
Alloc PC	Address of the system call that allocated the block.
what	Name of the process that owns the block, or "(fragment)" if the block is a fragment, or "(coalesced)" if the block was coalesced from adjacent free blocks.

show microcode

To display microcode image information available on line cards, use the **show microcode** command in EXEC mode.

show microcode

Syntax Description

This command has no arguments or keywords.

Command Modes

EXEC

Command History

Release	Modification
10.0	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Examples

The following is sample output from the **show microcode** command:

```
Router# show microcode

Microcode bundled in system

Card    Microcode    Target Hardware    Description
Type    Version      Version
----    ---------    ---------------    -----------
SP         2.3            11.x          SP version 2.3
EIP        1.1             1.x          EIP version 1.1
TRIP       1.2             1.x          TRIP version 1.2
FIP        1.4             2.x          FIP version 1.4
HIP        1.1             1.x          HIP version 1.1
SIP        1.1             1.x          SIP version 1.1
FSIP       1.1             1.x          FSIP version 1.1
```

In the following example for the Cisco 7200 series router, the output from the **show microcode** command lists the hardware types that support microcode download. For each type, the default microcode image name is displayed. If there is a configured default override, that name also is displayed.

```
router# show microcode

Microcode images for downloadable hardware
HW Type             Microcode image names
-----------------------------------------
ecpa     default    slot0:xcpa26-0
         configured slot0:xcpa26-2
pcpa     default    slot0:xcpa26-4
```

Related Commands

Command	Description
microcode (7000/7500)	Specifies where microcode should be loaded from on Cisco 7500/7000RSP routers.
microcode (7200)	Configures a default override for the microcode that is downloaded to the hardware on a Cisco 7200 series router.

show mls statistics

To display the Multilayer Switching (MLS) statistics for the Internet Protocol (IP), Internetwork Packet Exchange (IPX), multicast, Layer 2 protocol, and quality of service (QoS), use the **show mls statistics** command in user EXEC or privileged EXEC mode.

show mls statistics [**module** *num*]

Syntax Description

module *num*	(Optional) Displays the MLS statistics for a specific module.

Defaults

This command has no default settings.

Command Modes

User EXEC
Privileged EXEC

Command History

Release	Modification
12.2(14)SX	Support for this command was introduced on the Supervisor Engine 720.
12.2(17b)SXA	This command was changed to include the **module** *num* keyword and argument.
12.2(17d)SXB	Support for this command on the Supervisor Engine 2 was extended to Release 12.2(17d)SXB.
12.2(17d)SXB1	The output was changed to include total packets switched information.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

The total packets switched performance displayed is the rate calculated as the average rate in a period within the last 30 seconds.

The ingress ACL denied packet count is displayed in the Total packets L3 Switched field and in the Total packets dropped by ACL field.

The RPF failed packet count is displayed in the Total packets L3 Switched field.

If the IP multicast source sends traffic to any multicast group that does not have an (*,G) entry present in the mroute table, the **show mls statistics** command displays these packets as incrementing in the Total Mcast Packets Switched/Routed field. These packets are dropped in the hardware because there are no receivers for that group and no entry in the mroute table.

Examples

This example shows how to display the MLS statistics for all modules:

```
Router# show mls statistics

Statistics for Earl in Module 2

L2 Forwarding Engine
  Total packets Switched                : 20273@ 22552 pps
```

```
L3 Forwarding Engine
  Total Packets Bridged                 : 20273
  Total Packets FIB Switched            : 7864
  Total Packets ACL Routed              : 0
  Total Packets Netflow Switched        : 0
  Total Mcast Packets Switched/Routed   : 220598
  Total ip packets with TOS changed     : 0
  Total ip packets with COS changed     : 0
  Total non ip packets COS changed      : 0
  Total packets dropped by ACL          : 0
  Total packets dropped by Policing     : 705757744

Statistics for Earl in Module 9

L2 Forwarding Engine
  Total packets Switched                : 16683@ 1 pps

L3 Forwarding Engine
  Total Packets Bridged                 : 0
  Total Packets FIB Switched            : 0
  Total Packets ACL Routed              : 0
  Total Packets Netflow Switched        : 0
  Total Mcast Packets Switched/Routed   : 0
  Total ip packets with TOS changed     : 0
  Total ip packets with COS changed     : 0
  Total non ip packets COS changed      : 0
  Total packets dropped by ACL          : 0
  Total packets dropped by Policing     : 277949053
Router#
```

This example shows how to display the MLS statistics for a specific module:

```
Router# show mls statistics module 1

Statistics for Earl in Module 1

L2 Forwarding Engine
  Total packets Switched                : 2748166@ 22332 pps
>>
L3 Forwarding Engine
  Total Packets Bridged                 : 92750@ 34 pps
  Total Packets FIB Switched            : 7
  Total Packets ACL Routed              : 0
  Total Packets Netflow Switched        : 0
  Total Mcast Packets Switched/Routed   : 3079200
  Total ip packets with TOS changed     : 0
  Total ip packets with COS changed     : 0
  Total non ip packets COS changed      : 0
  Total packets dropped by ACL          : 0
  Total packets dropped by Policing     : 0
  Total Unicast RPF failed packets      : 0

Errors
  MAC/IP length inconsistencies         : 0
  Short IP packets received             : 0
  IP header checksum errors             : 0
  MAC/IPX length inconsistencies        : 0
  Short IPX packets received            : 0
Router#
```

Related Commands

Command	Description
show mls asic	display the application-specific integrated circuit (ASIC) version
show mls df-table	Displays information about the DF table.
show mls ip	Displays the Multilayer Switching (MLS) IP information.
show mls ipx	Displays the Multilayer Switching (MLS) IPX information.
show mls qos	Displays Multilayer Switching (MLS) quality of service (QoS) information
show mls statistics	Displays the Multilayer Switching (MLS) statistics for the Internet Protocol (IP)

show module

To display the module status and information, use the **show module** command in user EXEC or privileged EXEC mode.

show module [*mod-num* | **all** | **provision** | **version**]

Syntax Description

mod-num	(Optional) Number of the module.
all	(Optional) Displays the information for all modules.
provision	(Optional) Displays the status about the module provisioning.
version	(Optional) Displays the version information.

Defaults

This command has no default settings.

Command Modes

User EXEC
Privileged EXEC

Command History

Release	Modification
12.2(14)SX	Support for this command was introduced on the Supervisor Engine 720.
12.2(17d)SXB	Support for this command on the Supervisor Engine 2 was extended to Release 12.2(17d)SXB.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

In the Mod Sub-Module fields, the **show module** command displays the supervisor engine number but appends the uplink daughter card's module type and information.

Entering the **show module** command with no arguments is the same as entering the **show module all** command.

Examples

This example shows how to display information for all modules on a Cisco 7600 series router that is configured with a Supervisor Engine 720:

```
Router# show module

Mod Ports Card Type                                Model              Serial No.
--- ----- ---------------------------------------- ------------------ -----------
  5     2 Supervisor Engine 720 (Active)         WS-SUP720-BASE     SAD0644030K
  8    48 aCEF720 48 port 10/100/1000 Ethernet   WS-X6748-GE-TX     SAD07010045
  9    32 dCEF720 32 port Gigabit Ethernet       WS-X6832-SFP       SAD07010045

Mod MAC addresses                       Hw    Fw           Sw           Status
--- ---------------------------------- ------ ------------ ------------ -------
  5 00e0.aabb.cc00 to 00e0.aabb.cc3f   1.0    12.2(2003012 12.2(2003012 Ok
  8 0005.9a3b.d8c4 to 0005.9a3b.d8c7   0.705  7.1(0.12-Eng 12.2(2003012 Ok
  9 00e0.b0ff.f0f4 to 00e0.b0ff.f0f5   0.207  12.2(2002082 12.2(2003012 Ok
```

```
Mod Sub-Module                  Model           Serial           Hw     Status
--- --------------------------- --------------- ---------------- ------- -------
  5 Policy Feature Card 3       WS-F6K-PFC3     SAD0644031P      0.302   Ok
  5 MSFC3 Daughtercard          WS-SUP720       SAD06460172      0.701

Mod Online Diag Status
--- -------------------
  5 Not Available
  7 Bypass
  8 Bypass
  9 Bypass
Router#
```

This example shows how to display information for a specific module:

```
Router# show module 2

Mod Ports Card Type                                Model              Serial No.
--- ----- ---------------------------------------- ------------------ -----------
  5     2 Supervisor Engine 720 (Active)           WS-SUP720-BASE     SAD0644030K

Mod MAC addresses                          Hw    Fw           Sw           Status
--- ---------------------------------- ------ ------------ ------------ -------
  5 00e0.aabb.cc00 to 00e0.aabb.cc3f   1.0    12.2(2003012 12.2(2003012 Ok

Mod Sub-Module                  Model           Serial           Hw     Status
--- --------------------------- --------------- ---------------- ------- -------
  5 Policy Feature Card 3       WS-F6K-PFC3     SAD0644031P      0.302   Ok
  5 MSFC3 Daughtercard          WS-SUP720       SAD06460172      0.701

Mod Online Diag Status
--- -------------------
  5 Not Available
Router#
```

This example shows how to display version information:

```
Router# show module version

Mod Port Model              Serial #    Versions
--- ---- ------------------ ----------- -------------------------------------
  2 0    WS-X6182-2PA                   Hw : 1.0
                    Fw : 12.2(20030125:231135)
                    Sw : 12.2(20030125:231135)
  4 16   WS-X6816-GBIC      SAD04400CEE Hw : 0.205
         WS-F6K-DFC3A       SAD0641029Y Hw : 0.501
                    Fw : 12.2(20020828:202911)
                    Sw : 12.2(20030125:231135)
  6 2    WS-X6K-SUP3-BASE   SAD064300GU Hw : 0.705
                    Fw : 7.1(0.12-Eng-02)TAM
                    Sw : 12.2(20030125:231135)
                    Sw1: 8.1(0.45)KIS
         WS-X6K-SUP3-PFC3   SAD064200VR Hw : 0.701
                    Fw : 12.2(20021016:001154)
                    Sw : 12.2(20030125:231135)
         WS-F6K-PFC3        SAD064300M7 Hw : 0.301
  9 48   WS-X6548-RJ-45     SAD04490BAC Hw : 0.301
                    Fw : 6.3(1)
                    Sw : 7.5(0.30)CFW11
Router#
```

This example shows how to display module provisioning information:

```
Router# show module provision
```

```
Module Provision
  1    dynamic
  2    dynamic
  3    dynamic
  4    dynamic
  5    dynamic
  6    dynamic
  7    dynamic
  8    dynamic
  9    dynamic
 10    dynamic
 11    dynamic
 12    dynamic
 13    dynamic
Router#
```

Related Commands

Command	Description
show interfaces	Displays the status and statistics for the interfaces in the chassis.
show environment alarm	Displays the information about the environmental alarm.
show fm summary	Displays a summary of FM Information.
show environment status	Displays the information about the operational FRU status.

show monitor event-trace

To display event trace messages for Cisco IOS software subsystem components, use the **show monitor event-trace** command in privileged EXEC mode.

show monitor event-trace [**all-traces**] [*component* {**all** | **back** *hour:minute* | **clock** *hour:minute* | **from-boot** *seconds* | **latest** | **parameters**}]

Syntax Description

all-traces	(Optional) Displays all event trace messages in memory to the console.
component	(Optional) Name of the Cisco IOS software subsystem component that is the object of the event trace. To get a list of components that support event tracing in this release, use the **monitor event-trace ?** command.
all	Displays all event trace messages currently in memory for the specified component.
back *hour:minute*	Specifies how far back from the current time you want to view messages. For example, you can gather messages from the last 30 minutes. The time argument is specified in hours and minutes format (hh:mm).
clock *hour:minute*	Displays event trace messages starting from a specific clock time in hours and minutes format (hh:mm).
from-boot *seconds*	Displays event trace messages starting from a specified number of seconds after booting (uptime). To display the uptime, in seconds, enter the **show monitor event-trace** *component* **from-boot ?** command.
latest	Displays only the event trace messages since the last **show monitor event-trace** command was entered.
parameters	Displays the trace parameters. The only parameter displayed is the size (number of trace messages) of the trace file.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.0(18)S	This command was introduced.
12.2(8)T	This command was integrated into Cisco IOS Release 12.2(8)T.
12.2(25)S	This command was integrated into Cisco IOS Release 12.2(25)S. The **show monitor event-trace cef** comand replaced the **show cef events** and **show ip cef events** commands.
12.2(18)SXE	This command was integrated into Cisco IOS Release 12.2(18)SXE. The **spa** component keyword was added to support online insertion and removal (OIR) event messages for shared port adapters (SPAs). The **bfd** keyword was added for the *component* argument to display trace messages relating to the Bidirectional Forwarding Detection (BFD) feature.
12.4(4)T	Support for the **bfd** keyword was added for Cisco IOS Release 12.4(4)T.
12.0(31)S	Support for the **bfd** keyword was added for Cisco IOS Release 12.0(31)S.

Release	Modification
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB and implemented on the Cisco 10000 series routers.
12.4(9)T	The **cfd** keyword was added as an entry for the *component* argument to display trace messages relating to crypto fault detection.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.
12.2(33)SB	This command was integrated into Cisco IOS Release 12.2(33)SB.
12.4(20)T	This command was integrated into Cisco IOS Release 12.4(20)T.

Usage Guidelines

Use the **show monitor event-trace** command to display trace message information.

The trace function is not locked while information is being displayed to the console, which means that new trace messages can accumulate in memory. If entries accumulate faster than they can be displayed, some messages can be lost. If this happens, the **show monitor event-trace** command will generate a message indicating that some messages might be lost; however, messages will continue to display on the console. If the number of lost messages is excessive, the **show monitor event-trace** command will stop displaying messages.

Use the **bfd** keyword for the *component* argument to display trace messages relating to the BFD feature.

Use the **cfd** keyword for the *component* argument to display trace messages relating to the crypto fault detection feature. This keyword displays the contents of the error trace buffers in an encryption data path.

Examples

IPC Component Example

The following is sample output from the **show monitor event-trace** *component* command for the interprocess communication (IPC) component. Notice that each trace message is numbered and is followed by a time stamp (derived from the device uptime). Following the time stamp is the component-specific message data.

```
Router# show monitor event-trace ipc

3667:  6840.016:Message type:3 Data=0123456789
3668:  6840.016:Message type:4 Data=0123456789
3669:  6841.016:Message type:5 Data=0123456789
3670:  6841.016:Message type:6 Data=0123456
```

BFD Component for Cisco IOS Release 12.2(18)SXE, 12.0(31)S, and 12.4(4)T

Use the **show monitor event-trace bfd all** command to display logged messages for important BFD events in the recent past. The following trace messages show BFD session state changes:

```
Router# show monitor event-trace bfd all

    3d03h: EVENT: Session [172.16.10.2,172.16.10.1,Fa6/0,1], event Session
           create, state Unknown -> Fail
    3d03h: EVENT: Session [172.16.10.2,172.16.10.1,Fa6/0,1], state Fail -> Down
            (from LC)
    3d03h: EVENT: Session [172.16.10.2,172.16.10.1,Fa6/0,1], state Down -> Init
            (from LC)
    3d03h: EVENT: Session [172.16.10.2,172.16.10.1,Fa6/0,1], state Init -> Up
           (from LC)
    3d07h: EVENT: Session [172.16.10.2,172.16.10.1,Fa6/0,2], event Session
           create, state Unknown -> Fail
```

```
3d07h: EVENT: Session [172.16.10.2,172.16.10.1,Fa6/0,2], state Fail -> Down
        (from LC)
3d07h: EVENT: Session [172.16.10.2,172.16.10.1,Fa6/0,2], state Down -> Up
       (from LC)
```

To display trace information for all components configured for event tracing on the networking device, enter the **show monitor event-trace all-traces** command. In this example, separate output is provided for each event, and message numbers are interleaved between the events.

```
Router# show monitor event-trace all-traces

Test1 event trace:
3667: 6840.016:Message type:3 Data=0123456789
3669: 6841.016:Message type:4 Data=0123456789
3671: 6842.016:Message type:5 Data=0123456789
3673: 6843.016:Message type:6 Data=0123456789

Test2 event trace:
3668: 6840.016:Message type:3 Data=0123456789
3670: 6841.016:Message type:4 Data=0123456789
3672: 6842.016:Message type:5 Data=0123456789
3674: 6843.016:Message type:6 Data=0123456789
```

SPA Component Example

The following is sample output from the **show monitor event-trace** *component* **latest** command for the **spa** component:

```
Router# show monitor event-trace spa latest

00:01:15.364: subslot 2/3: 4xOC3 POS SPA, TSM Event:inserted  New state:wait_psm
_ready
     spa type 0x440
00:02:02.308: subslot 2/0: not present, TSM Event:empty  New state:remove
     spa type 0x0, fail code 0x0(none)
00:02:02.308: subslot 2/0: not present, TSM Event:remove_complete  New state:idle
00:02:02.308: subslot 2/1: not present, TSM Event:empty  New state:remove
     spa type 0x0, fail code 0x0(none)
00:02:02.308: subslot 2/1: not present, TSM Event:remove_complete  New state:idle
00:02:02.308: subslot 2/2: not present, TSM Event:empty  New state:remove
     spa type 0x0, fail code 0x0(none)
00:02:02.308: subslot 2/2: not present, TSM Event:remove_complete  New state:idle
00:02:02.312: subslot 2/3: not present(plugin 4xOC3 POS SPA), TSM Event:empty  New
state:remove
     spa type 0x0, fail code 0x0(none)
00:02:02.312: subslot 2/3: not present, TSM Event:remove_complete  New state:idle
```

Cisco Express Forwarding Component Examples

If you select Cisco Express Forwarding as the component for which to display event messages, you can use the following additional arguments and keywords: **show monitor event-trace cef** [**events** | **interface** | **ipv6** | **ipv4**][**all**].

The following example shows the IPv6 or IPv4 events related to the Cisco Express Forwarding component. Each trace message is numbered and is followed by a time stamp (derived from the device uptime). Following the time stamp is the component-specific message data.

```
Router# show monitor event-trace cef ipv6 all

00:00:24.612:  [Default] *::*/*'00             New FIB table          [OK]

Router# show monitor event-trace cef ipv4 all

00:00:24.244:  [Default] 127.0.0.81/32'01       FIB insert             [OK]
```

In the following example, all event trace messages for the Cisco Express Forwarding component are displayed:

```
Router# show monitor event-trace cef events all

00:00:18.884: SubSys  fib_ios_chain init
00:00:18.884: Inst    unknown -> RP
00:00:24.584: SubSys  fib init
00:00:24.592: SubSys  fib_ios init
00:00:24.592: SubSys  fib_ios_if init
00:00:24.596: SubSys  ipv4fib init
00:00:24.608: SubSys  ipv4fib_ios init
00:00:24.612: SubSys  ipv6fib_ios init
00:00:24.620: Flag    IPv4 CEF enabled set to yes
00:00:24.620: Flag    0x7BF6B62C set to yes
00:00:24.620: Flag    IPv4 CEF switching enabled set to yes
00:00:24.624: GState  CEF enabled
00:00:24.628: SubSys  ipv4fib_les init
00:00:24.628: SubSys  ipv4fib_pas init
00:00:24.632: SubSys  ipv4fib_util init
00:00:25.304: Process Background created
00:00:25.304: Flag    IPv4 CEF running set to yes
00:00:25.304: Process Background event loop enter
00:00:25.308: Flag    IPv4 CEF switching running set to yes
```

The following example shows Cisco Express Forwarding interface events:

```
Router# show monitor event-trace cef interface all

00:00:24.624: <empty>      (sw  4) Create   new
00:00:24.624: <empty>      (sw  4) SWIDBLnk FastEthernet0/0(4)
00:00:24.624: Fa0/0        (sw  4) NameSet
00:00:24.624: <empty>      (hw  1) Create   new
00:00:24.624: <empty>      (hw  1) HWIDBLnk FastEthernet0/0(1)
00:00:24.624: Fa0/0        (hw  1) NameSet
00:00:24.624: <empty>      (sw  3) Create   new
00:00:24.624: <empty>      (sw  3) SWIDBLnk FastEthernet0/1(3)
00:00:24.624: Fa0/1        (sw  3) NameSet
00:00:24.624: <empty>      (hw  2) Create   new
```

Cisco Express Forwarding Component Examples for Cisco 10000 Series Routers Only

The following example shows the IPv4 events related to the Cisco Express Forwarding component. Each trace message is numbered and is followed by a time stamp (derived from the device uptime). Following the time stamp is the component-specific message data.

```
Router# show monitor event-trace cef ipv4 all

00:00:48.244:  [Default] 127.0.0.81/32'01      FIB insert               [OK]
```

In the following example, all event trace message for the Cisco Express Forwarding component are displayed:

```
Router# show monitor event-trace cef events all

00:00:18.884: SubSys  fib_ios_chain init
00:00:18.884: Inst    unknown -> RP
00:00:24.584: SubSys  fib init
00:00:24.592: SubSys  fib_ios init
00:00:24.592: SubSys  fib_ios_if init
00:00:24.596: SubSys  ipv4fib init
00:00:24.608: SubSys  ipv4fib_ios init
00:00:24.620: Flag    IPv4 CEF enabled set to yes
```

```
00:00:24.620: Flag    0x7BF6B62C set to yes
00:00:24.620: Flag    IPv4 CEF switching enabled set to yes
00:00:24.624: GState  CEF enabled
00:00:24.628: SubSys  ipv4fib_les init
00:00:24.628: SubSys  ipv4fib_pas init
00:00:24.632: SubSys  ipv4fib_util init
00:00:25.304: Process Background created
00:00:25.304: Flag    IPv4 CEF running set to yes
00:00:25.304: Process Background event loop enter
00:00:25.308: Flag    IPv4 CEF switching running set to yes
```

The following examples show Cisco Express Forwarding interface events:

```
Router# show monitor event-trace cef interface all

00:00:24.624: <empty>      (sw  4) Create   new
00:00:24.624: <empty>      (sw  4) SWIDBLnk FastEthernet1/0/0(4)
00:00:24.624: Fa0/0        (sw  4) NameSet
00:00:24.624: <empty>      (hw  1) Create   new
00:00:24.624: <empty>      (hw  1) HWIDBLnk FastEthernet1/0/0(1)
00:00:24.624: Fa0/0        (hw  1) NameSet
00:00:24.624: <empty>      (sw  3) Create   new
00:00:24.624: <empty>      (sw  3) SWIDBLnk FastEthernet1/1/0(3)
00:00:24.624: Fa0/1        (sw  3) NameSet
00:00:24.624: <empty>      (hw  2) Create   new
```

CFD Component for Cisco IOS Release 12.4(9)T

To troubleshoot errors in an encryption datapath, enter the **show monitor event-trace cfd all** command. In this example, events are shown separately, each beginning with a time stamp, followed by data from the error trace buffer. Cisco Technical Assistence Center (TAC) engineers can use this information to diagnose the cause of the errors.

Note If no packets have been dropped, this command does not display any output.

```
Router# show monitor event-trace cfd all

00:00:42.452: 450000B4 00060000 FF33B306 02020203 02020204 32040000 F672999C
        00000001 7A7690C2 A0A4F8BC E732985C D6FFDCC8 00000001 C0902BD0
        A99127AE 8EAA22D4

00:00:44.452: 450000B4 00070000 FF33B305 02020203 02020204 32040000 F672999C
        00000002 93C01218 2325B697 3C384CF1 D6FFDCC8 00000002 BFA13E8A
        D21053ED 0F62AB0E

00:00:46.452: 450000B4 00080000 FF33B304 02020203 02020204 32040000 F672999C
        00000003 7D2E11B7 A0BA4110 CC62F91E D6FFDCC8 00000003 7236B930
        3240CA8C 9EBB44FF

00:00:48.452: 450000B4 00090000 FF33B303 02020203 02020204 32040000 F672999C
        00000004 FB6C80D9 1AADF938 CDE57ABA D6FFDCC8 00000004 E10D8028
        6BBD748F 87F5E253

00:00:50.452: 450000B4 000A0000 FF33B302 02020203 02020204 32040000 F672999C
        00000005 697C8D9D 35A8799A 2A67E97B D6FFDCC8 00000005 BC21669D
        98B29FFF F32670F6

00:00:52.452: 450000B4 000B0000 FF33B301 02020203 02020204 32040000 F672999C
        00000006 CA18CBC4 0F387FE0 9095C27C D6FFDCC8 00000006 87A54811
        AE3A0517 F8AC4E64
```

Related Commands

Command	Description
monitor event-trace (EXEC)	Controls event trace functions for a specified Cisco IOS software subsystem component.
monitor event-trace (global)	Configures event tracing for a specified Cisco IOS software subsystem component.
monitor event-trace dump-traces	Saves trace messages for all event traces currently enabled on the networking device.

show monitor permit-list

To display the permit-list state and interfaces configured, use the **show monitor permit-list** command in user EXEC or privileged EXEC mode.

show monitor permit-list

Syntax Description

This command has no arguments or keywords.

Defaults

This command has no default settings.

Command Modes

User EXEC
Privileged EXEC

Command History

Release	Modification
12.2(18)SXE	Support for this command was introduced on the Supervisor Engine 720.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Examples

This example shows how to display the permit-list state and interfaces configured:

```
Router# show monitor permit-list

SPAN Permit-list      :Admin Enabled
 Permit-list ports     :Gi5/1-4,Gi6/1
Router(config)#
```

Related Commands

Command	Description
monitor permit-list	Configures a destination port permit list or adds to an existing destination port permit list.

show monitor session

To display information about the ERSPAN, SPAN and RSPAN sessions, use the **show monitor session** command in user EXEC mode.

show monitor session [**range** *session-range* | **local** | **remote** | **all** | *session*]

show monitor session [**erspan-destination** | **erspan-source** | **egress replication-mode capability**| **detail**]

Syntax Description

range *session-range*	(Optional) Displays a range of sessions; valid values are from 1 to 66.
local	(Optional) Displays only local SPAN sessions.
remote	(Optional) Displays both RSPAN source and destination sessions.
all	(Optional) Displays all sessions.
session	(Optional) Number of the session; valid values are from 1 to 66.
erspan-destination	(Optional) Displays information about the destination ERSPAN sessions only. This keyword is not supported on the Supervisor Engine 2.
erspan-source	(Optional) Displays information about the source ERSPAN sessions only. This keyword is not supported on the Supervisor Engine 2.
egress replication-mode capability	(Optional) Displays the operational mode and configured mode of the session and module session capabilities.
detail	(Optional) Displays detailed session information.

Defaults

This command has no default settings.

Command Modes

User EXEC (>)

Command History

Release	Modification
12.2(14)SX	This command was introduced on the Supervisor Engine 720.
12.2(17d)SXB	Support was added for the Supervisor Engine 2.
12.2(18)SXE	Support was added for the **erspan-destination** and **erspan-source** keywords on the Supervisor Engine 720 only.
12.2(18)SXF	This command was updated as follows: • Support was added for the Supervisor Engine 32. • ERSPAN is supported in any switch fabric module functionality switching mode.
12.2(33)SXH	The **egress replication-mode capability** keywords were added.

Usage Guidelines

The **erspan-destination** and **erspan-source** keywords are not supported on Catalyst 6500 series switches that are configured with a Supervisor Engine 2.

In releases prior to Release 12.2(18)SXF, ERSPAN is supported on Catalyst 6500 series switches that are operating in compact switch fabric module functionality switching mode only.

Release 12.2(18)SXF and later releases support ERSPAN in any switch fabric module functionality switching mode.

If the switch fabric module functionality switching mode is set to compact, the output of the **show** commands display "dcef mode" for fabric-enabled modules with DFC3 installed and display "fabric mode" for other fabric-enabled modules.

If the switch fabric module functionality switching mode is set to truncated, the output of the **show** commands display "fabric mode" for all fabric-enabled modules.

When entering a range of sessions, use a dash (-) to specify a range and separate multiple entries with a comma (,). Do not enter spaces before or after the comma or the dash.

You can enter multiple ranges by separating the ranges with a comma.

If you enter the **show monitor session** command without specifying a session, the information for all sessions is displayed.

Examples

This example shows how to display the saved version of the monitor configuration for a specific session:

```
Router# show monitor session 2
Session 2
------------
Type : Remote Source Session

Source Ports:
    RX Only:       Fa1/1-3
Dest RSPAN VLAN:   901
Router#
```

This example shows how to display the detailed information from a saved version of the monitor configuration for a specific session:

```
Router# show monitor session 2 detail
Session 2
------------
Type : Remote Source Session

Source Ports:
    RX Only:       Fa1/1-3
    TX Only:       None
    Both:          None
Source VLANs:
    RX Only:       None
    TX Only:       None
    Both:          None
Source RSPAN VLAN: None
Destination Ports: None
Filter VLANs:      None
Dest RSPAN VLAN:   901
Router#
```

This example shows how to display information about the egress replication mode only:

```
Router# show monitor session egress replication-mode capability
 No SPAN configuration is present in the system.
```

```
-------------------------------------------------------
Global Egress SPAN Replication Mode Capability:

Slot          Egress Replication Capability
 No       LSPAN           RSPAN             ERSPAN
-------------------------------------------------------
 3     Distributed      Distributed        Distributed
 5     Distributed      Distributed        Distributed
Router#
```

This example shows how to display information about the destination ERSPAN sessions only:

```
Router# show monitor session erspan-destination
Session 2
---------
Type            : ERSPAN Destination Session
Status         : Admin Disabled
Router#
```

This example shows how to display detailed information about the destination ERSPAN sessions only:

```
Router# show monitor session erspan-destination detail
Session 2
---------
Type                   : ERSPAN Destination Session
Status                 : Admin Disabled
Description            : -
Source Ports           :
    RX Only            : None
    TX Only            : None
    Both               : None
Source VLANs           :
    RX Only            : None
    TX Only            : None
    Both               : None
Source RSPAN VLAN      : None
Destination Ports      : None
Filter VLANs           : None
Destination RSPAN VLAN : None
Source IP Address      : None
Source IP VRF          : None
Source ERSPAN ID       : None
Destination IP Address : None
Destination IP VRF     : None
Destination ERSPAN ID  : None
Origin IP Address      : None
IP QOS PREC            : 0
IP TTL                 : 255
Router#
```

This example shows how to display information about the source ERSPAN sessions only:

```
Router# show monitor session erspan-source
Session 1
---------
Type                   : ERSPAN Source Session
Status                : Admin Disabled
Session 3
---------
Type                   : ERSPAN Source Session
Status                : Admin Disabled
Router#
```

This example shows how to display detailed information about the source ERSPAN sessions only:

```
Router# show monitor session erspan-source detail
Session 1
---------
Type                  : ERSPAN Source Session
Status                : Admin Disabled
Description           : -
Source Ports          :
    RX Only           : None
    TX Only           : None
    Both              : None
Source VLANs          :
    RX Only           : None
    TX Only           : None
    Both              : None
Source RSPAN VLAN     : None
Destination Ports     : None
Filter VLANs          : None
Destination RSPAN VLAN : None
Source IP Address     : None
Source IP VRF         : None
Source ERSPAN ID      : None
Destination IP Address : None
Destination IP VRF    : None
Destination ERSPAN ID  : None
Origin IP Address     : None
IP QOS PREC           : 0
IP TTL                : 255

Session 3
---------
Type                  : ERSPAN Source Session
Status                : Admin Disabled
Description           : -
Source Ports          :
    RX Only           : None
    TX Only           : None
    Both              : None
Source VLANs          :
    RX Only           : None
    TX Only           : None
    Both              : None
Source RSPAN VLAN     : None
Destination Ports     : None
Filter VLANs          : None
Destination RSPAN VLAN : None
Source IP Address     : None
Source IP VRF         : None
Source ERSPAN ID      : None
Destination IP Address : None
Destination IP VRF    : None
Destination ERSPAN ID  : None
Origin IP Address     : None
IP QOS PREC           : 0
IP TTL                : 255
Router#
```

This example shows how to display the operational mode and configured mode of the session and module session capabilities:

```
Router# show monitor session egress replication-mode capability
Session 65 Type Local Session
----------------------------------------------
```

```
Operational mode of egress span replication    : Centralized
Configured mode of egress span replication     : Distributed/Default

 Slot            Egress Replication Capability
-------------------------------------------------
  1                   Centralized
  3                   Centralized
  5                   Centralized
Router#
```

Related Commands

Command	Description
monitor session	Starts a new ERSPAN, SPAN, or RSPAN session, adds or deletes interfaces or VLANs to or from an existing session, filters ERSPAN, SPAN, or RSPAN traffic to specific VLANs, or deletes a session.
monitor session type	Creates an ERSPAN source session number or enters the ERSPAN session configuration mode for the session.
remote-span	Configures a VLAN as an RSPAN VLAN.

show msfc

To display Multilayer Switching Feature Card (MSFC) information, use the **show msfc** command in user EXEC or privileged EXEC mode.

show msfc {**buffers** | **eeprom** | **fault** | **netint** | **tlb**}

Syntax Description

buffers	Displays buffer-allocation information.
eeprom	Displays the internal information.
fault	Displays fault information.
netint	Displays network-interrupt information.
tlb	Displays information about the TLB registers.

Defaults

This command has no default settings.

Command Modes

User EXEC
Privileged EXEC

Command History

Release	Modification
12.2(14)SX	Support for this command was introduced on the Supervisor Engine 720.
12.2(17d)SXB	Support for this command on the Supervisor Engine 2 was extended to Release 12.2(17d)SXB.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Examples

These examples display the **show msfc** command output:

```
Router# show msfc buffers

Reg. set    Min    Max
  TX               640
 ABQ         640  16384
   0           0     40
   1        6715   8192
   2           0      0
   3           0      0
   4           0      0
   5           0      0
   6           0      0
   7           0      0
Threshold = 8192

Vlan  Sel  Min  Max  Cnt  Rsvd
1016    1 6715 8192    0     0
Router#

Router# show msfc eeprom
```

```
RSFC CPU IDPROM:
IDPROM image:

  (FRU is 'Cat6k MSFC 2 daughterboard')

IDPROM image block #0:
  hexadecimal contents of block:
  00: AB AB 01 90 13 22 01 00 00 02 60 03 00 EA 43 69    ....."....`...Ci
  10: 73 63 6F 20 53 79 73 74 65 6D 73 00 00 00 00 00    sco Systems.....
  20: 00 00 57 53 2D 46 36 4B 2D 4D 53 46 43 32 00 00    ..WS-F6K-MSFC2..
  30: 00 00 00 00 00 00 53 41 44 30 36 32 31 30 30 36    ......SAD0621006
  40: 37 00 00 00 00 00 00 00 00 00 37 33 2D 37 32 33    7.........73-723
  50: 37 2D 30 33 00 00 00 00 00 00 41 30 00 00 00 00    7-03......A0....
  60: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
  70: 00 00 00 02 00 03 00 00 00 00 00 09 00 05 00 01    ................
  80: 00 03 00 01 00 01 00 02 00 EA FF DF 00 00 00 00    ................

  block-signature = 0xABAB, block-version = 1,
  block-length = 144, block-checksum = 4898

  *** common-block ***
  IDPROM capacity (bytes) = 256  IDPROM block-count = 2
  FRU type = (0x6003,234)
  OEM String = 'Cisco Systems'
  Product Number = 'WS-F6K-MSFC2'
  Serial Number = 'SAD06210067'
  Manufacturing Assembly Number = '73-7237-03'
  Manufacturing Assembly Revision = 'A0'
  Hardware Revision = 2.3
  Manufacturing bits = 0x0  Engineering bits = 0x0
  SNMP OID = 9.5.1.3.1.1.2.234
  Power Consumption = -33 centiamperes    RMA failure code = 0-0-0-0
  *** end of common block ***

IDPROM image block #1:
  hexadecimal contents of block:
  00: 60 03 01 62 0A C2 00 00 00 00 00 00 00 00 00 00    `..b............
  10: 00 00 00 00 00 00 01 00 23 00 08 7C A4 CE 80 00 40    .......#..|....@
  20: 01 01 00 01 00 00 00 00 00 00 00 00 00 00 00 00    ................
  30: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
  40: 14 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
  50: 10 00 4B 3C 41 32 80 80 80 80 80 80 80 80 80 80    ..K<A2..........
  60: 80 80                                              ..

  block-signature = 0x6003, block-version = 1,
  block-length = 98, block-checksum = 2754

  *** linecard specific block ***
  feature-bits =   00000000 00000000
  hardware-changes-bits =   00000000 00000001
  card index = 35
  mac base = 0008.7CA4.CE80
  mac_len = 64
  num_processors = 1
  epld_num = 1
  epld_versions = 0001 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 00
00 0000 0000
  port numbers:
    pair #0: type=14, count=01
    pair #1: type=00, count=00
    pair #2: type=00, count=00
    pair #3: type=00, count=00
    pair #4: type=00, count=00
    pair #5: type=00, count=00
```

```
    pair #6: type=00, count=00
    pair #7: type=00, count=00
  sram_size = 4096
  sensor_thresholds =
    sensor #0: critical = 75 oC, warning = 60 oC
    sensor #1: critical = 65 oC, warning = 50 oC
    sensor #2: critical = -128 oC (sensor not present), warning = -128 oC (senso
r not present)
    sensor #3: critical = -128 oC (sensor not present), warning = -128 oC (senso
r not present)
    sensor #4: critical = -128 oC (sensor not present), warning = -128 oC (senso
r not present)
    sensor #5: critical = -128 oC (sensor not present), warning = -128 oC (senso
r not present)
    sensor #6: critical = -128 oC (sensor not present), warning = -128 oC (senso
r not present)
    sensor #7: critical = -128 oC (sensor not present), warning = -128 oC (senso
r not present)
  *** end of linecard specific block ***

End of IDPROM image
Router#

Router# show msfc fault

 Reg. set    Min    Max
  TX               640
 ABQ         640  16384
   0           0     40
   1        6715   8192
   2           0      0
   3           0      0
   4           0      0
   5           0      0
   6           0      0
   7           0      0
Threshold = 8192

Vlan  Sel  Min  Max  Cnt  Rsvd
1016    1 6715 8192    0     0
Router#

Router# show msfc netint

Network IO Interrupt Throttling:
 throttle count=0, timer count=0
 active=0, configured=1
 netint usec=3999, netint mask usec=400

Router#

Router# show msfc tlb

Mistral revision 3
TLB entries : 37
Virt Address range      Phy Address range      Attributes
0x10000000:0x1001FFFF   0x010000000:0x01001FFFF   CacheMode=2, RW, Valid
0x10020000:0x1003FFFF   0x010020000:0x01003FFFF   CacheMode=2, RW, Valid
0x10040000:0x1005FFFF   0x010040000:0x01005FFFF   CacheMode=2, RW, Valid
0x10060000:0x1007FFFF   0x010060000:0x01007FFFF   CacheMode=2, RW, Valid
0x10080000:0x10087FFF   0x010080000:0x010087FFF   CacheMode=2, RW, Valid
0x10088000:0x1008FFFF   0x010088000:0x01008FFFF   CacheMode=2, RW, Valid
0x18000000:0x1801FFFF   0x010000000:0x01001FFFF   CacheMode=0, RW, Valid
0x19000000:0x1901FFFF   0x010000000:0x01001FFFF   CacheMode=7, RW, Valid
```

```
0x1E000000:0x1E1FFFFF   0x01E000000:0x01E1FFFFF   CacheMode=2, RW, Valid
0x1E880000:0x1E881FFF   0x01E880000:0x01E881FFF   CacheMode=2, RW, Valid
0x1FC00000:0x1FC7FFFF   0x01FC00000:0x01FC7FFFF   CacheMode=2, RO, Valid
0x30000000:0x3001FFFF   0x070000000:0x07001FFFF   CacheMode=2, RW, Valid
0x40000000:0x407FFFFF   0x000000000:0x0007FFFFF   CacheMode=3, RO, Valid
0x40800000:0x40FFFFFF   0x000800000:0x000FFFFFF   CacheMode=3, RO, Valid
0x41000000:0x417FFFFF   0x001000000:0x0017FFFFF   CacheMode=3, RO, Valid
0x41800000:0x419FFFFF   0x001800000:0x0019FFFFF   CacheMode=3, RO, Valid
0x41A00000:0x41A7FFFF   0x001A00000:0x001A7FFFF   CacheMode=3, RO, Valid
0x41A80000:0x41A9FFFF   0x001A80000:0x001A9FFFF   CacheMode=3, RO, Valid
0x41AA0000:0x41ABFFFF   0x001AA0000:0x001ABFFFF   CacheMode=3, RO, Valid
0x41AC0000:0x41AC7FFF   0x001AC0000:0x001AC7FFF   CacheMode=3, RO, Valid
0x41AC8000:0x41ACFFFF   0x001AC8000:0x001ACFFFF   CacheMode=3, RO, Valid
0x41AD0000:0x41AD7FFF   0x001AD0000:0x001AD7FFF   CacheMode=3, RO, Valid
0x41AD8000:0x41AD9FFF   0x001AD8000:0x001AD9FFF   CacheMode=3, RO, Valid
0x41ADA000:0x41ADBFFF   0x001ADA000:0x001ADBFFF   CacheMode=3, RW, Valid
0x41ADC000:0x41ADDFFF   0x001ADC000:0x001ADDFFF   CacheMode=3, RW, Valid
0x41ADE000:0x41ADFFFF   0x001ADE000:0x001ADFFFF   CacheMode=3, RW, Valid
0x41AE0000:0x41AFFFFF   0x001AE0000:0x001AFFFFF   CacheMode=3, RW, Valid
0x41B00000:0x41B7FFFF   0x001B00000:0x001B7FFFF   CacheMode=3, RW, Valid
0x41B80000:0x41BFFFFF   0x001B80000:0x001BFFFFF   CacheMode=3, RW, Valid
0x41C00000:0x41DFFFFF   0x001C00000:0x001DFFFFF   CacheMode=3, RW, Valid
0x41E00000:0x41FFFFFF   0x001E00000:0x001FFFFFF   CacheMode=3, RW, Valid
0x42000000:0x43FFFFFF   0x002000000:0x003FFFFFF   CacheMode=3, RW, Valid
0x44000000:0x45FFFFFF   0x004000000:0x005FFFFFF   CacheMode=3, RW, Valid
0x46000000:0x47FFFFFF   0x006000000:0x007FFFFFF   CacheMode=3, RW, Valid
0x06E00000:0x06FFFFFF   0x006E00000:0x006FFFFFF   CacheMode=2, RW, Valid
0x07000000:0x077FFFFF   0x007000000:0x0077FFFFF   CacheMode=2, RW, Valid
0x07800000:0x07FFFFFF   0x007800000:0x007FFFFFF   CacheMode=2, RW, Valid

Router#
```

Related Commands

Command	Description
show environment alarm	Displays the information about the environmental alarm.
show fm summary	Displays a summary of FM Information.
show environment status	Displays the information about the operational FRU status.

show pagp

To display port-channel information, use the **show pagp** command in user EXEC or privileged EXEC mode.

show pagp [*group-number*] {**counters** | **internal** | **neighbor** | **pgroup**}

Syntax Description

group-number	(Optional) Channel-group number; valid values are a maximum of 64 values from 1 to 282.
counters	Displays the traffic information.
internal	Displays the internal information.
neighbor	Displays the neighbor information.
pgroup	Displays the active port channels.

Defaults

This command has no default settings.

Command Modes

User EXEC
Privileged EXEC

Command History

Release	Modification
12.2(14)SX	Support for this command was introduced on the Supervisor Engine 720.
12.2(17d)SXB	Support for this command on the Supervisor Engine 2 was extended to Release 12.2(17d)SXB.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

You can enter any **show pagp** command to display the active port-channel information. To display the nonactive information, enter the **show pagp** command with a group.

The **port-channel** *number* values from 257 to 282 are supported on the CSM and the FWSM only.

Examples

This example shows how to display information about the PAgP counters:

```
Router# show pagp counters

          Information        Flush
Port      Sent   Recv     Sent   Recv
---------------------------------------
Channel group: 1
  Fa5/4    2660   2452     0      0
  Fa5/5    2676   2453     0      0
Channel group: 2
  Fa5/6    289    261      0      0
  Fa5/7    290    261      0      0
Channel group: 1023
  Fa5/9    0      0        0      0
```

```
Channel group: 1024
  Fa5/8    0      0        0      0
Router#
```

This example shows how to display internal PAgP information:

```
Router# show pagp 1 internal

Flags:  S - Device is sending Slow hello.  C - Device is in Consistent state.
        A - Device is in Auto mode.
Timers: H - Hello timer is running.        Q - Quit timer is running.
        S - Switching timer is running.    I - Interface timer is running.

Channel group 1
                              Hello    Partner  PAgP     Learning
Port      Flags State   Timers  Interval Count   Priority  Method
Fa5/4     SC    U6/S7           30s      1        128       Any
Fa5/5     SC    U6/S7           30s      1        128       Any
Router#
```

This example shows how to display PAgP-neighbor information for all neighbors:

```
Router# show pagp neighbor

Flags:  S - Device is sending Slow hello.  C - Device is in Consistent state.
        A - Device is in Auto mode.        P - Device learns on physical port.

Channel group 1 neighbors
          Partner              Partner          Partner          Partner Group
Port      Name                 Device ID        Port       Age   Flags   Cap.
Fa5/4     JAB031301            0050.0f10.230c   2/45         2s  SAC     2D
Fa5/5     JAB031301            0050.0f10.230c   2/46        27s  SAC     2D

Channel group 2 neighbors
          Partner              Partner          Partner          Partner Group
Port      Name                 Device ID        Port       Age   Flags   Cap.
Fa5/6     JAB031301            0050.0f10.230c   2/47        10s  SAC     2F
Fa5/7     JAB031301            0050.0f10.230c   2/48        11s  SAC     2F

Channel group 1023 neighbors
          Partner              Partner          Partner          Partner Group
Port      Name                 Device ID        Port       Age   Flags   Cap.

Channel group 1024 neighbors
          Partner              Partner          Partner          Partner Group
Port      Name                 Device ID        Port       Age   Flags   Cap.
Router#
```

Related Commands

Command	Description
pagp learn-method	Learns the input interface of the incoming packets.
pagp port-priority	Selects a port in hot standby mode.

show parser dump

To display the CLI syntax options for all command modes or for a specified command mode, use the **show parser dump** command in privileged EXEC mode.

show parser dump {*command-mode* | **all**} [**privilege-level** *level*] [**extended**] [**breakage**]

Syntax Description

command-mode	A keyword indicating the command mode. The output will include the syntax for commands only in the specified command mode. The list of command mode keywords will vary depending on your software image. Use the **show parser dump ?** command to display the list of command mode keyword options. For further assistance determining the proper command mode, see the "Cisco IOS Command Modes" Release 12.2 document, available on Cisco.com.
all	Indicates that all commands in all modes should be displayed in the output. **Caution** This keyword generates a very large amount of output, which may exceed your system or buffer memory.
privilege-level *level*	(Optional) Lists CLI commands only with the privilege level specified in the level argument.
breakage	(Optional) Enables detection of potential parser chain syntax breakage. This keyword is intended for internal use.
extend	(Optional) Enables the extended display mode. The extended parser display shows the keyword and argument descriptions typically shown with the command-line help (**?** command). **Note** This keyword can produce a large amount of output.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(4)T	This command was introduced.
12.2(13)T, 12.0(23)S	This command was enhanced to resolve certain execution errors.

Usage Guidelines

This command was developed to allow the exploration of the CLI command syntax without requiring the user to actually enter a specific mode and use the **?** command line help.

Caution Use caution when entering this command with the **all** keyword. A large amount of output can be generated by this command, which may easily exceed buffer or system memory on smaller platforms. Also, some configuration modes have hundreds of valid commands. For large dumps, use of the

redirection to a file using the **| redirect** *URL* syntax at the end of the command is highly recommended. (See the documentation for the **show <command> redirect** command for more information on using this command extension.)

Output for this command will show the syntax options for all commands available in the specified mode. The preceding number shows the privilege level associated with that command. For example, the line

```
15 type dhcp
```

indicates that the **type dhcp** command has a privilege level of 15 assigned to it. For information about privilege levels, see the "Configuring Passwords and Privileges" chapter in the *Cisco IOS Security Configuration Guide*.

Any given command-line string should indicate the full syntax needed to make the command complete and valid. In other words, the command line string ends where the carriage return (Enter) could be entered, as indicated in command-line help by the <cr> syntax. You will typically see multiple forms of a command, each showing a valid syntax combination. For example, each of the following syntax combinations, as seen in the output of the **show parser dump rtr | include dhcp** command, are valid commands:

```
type dhcp dest-ipaddr <address> source-ipaddr <address> option <82-82> circuit-id <string>
type dhcp dest-ipaddr <address> source-ipaddr <address> option <82-82> remote-id <string>
type dhcp dest-ipaddr <address> source-ipaddr <address> option <82-82> subnet-mask
<ipmask>
type dhcp dest-ipaddr <address> source-ipaddr <address> option <82-82>
type dhcp dest-ipaddr <address> source-ipaddr <address>
type dhcp dest-ipaddr <address>
type dhcp
```

Use of the show command extensions **| begin**, **| include**, and **| exclude** are recommended for this command, as these extensions allow you to filter the output to show only the commands you are interested in. The redirection extensions **| redirect**, **| append**, and **| tee** allow you to redirect the output of this command to local or remote storage as a file.

As with most **show** commands, you can typically exit from the --More-- prompt back to EXEC mode using Ctrl-Z. For some connections, Ctrl-Shift-6 (Ctrl^) or Ctrl-Shift-6-X should be used instead.

Examples

The following example shows a typical list of command mode keywords:

```
Router# show parser dump ?
  aaa-user                  AAA user definition
  accept-dialin             VPDN group accept dialin configuration mode
  accept-dialout            VPDN group accept dialout configuration mode
  address-family            Address Family configuration mode
  aic                       Alarm Interface Card configuration mode
  all                       For all modes
  bba-group                 BBA Group configuration mode
  bsm-cfg                   BSM config definition
  cascustom                 Cas custom configuration mode
  clid-group                CLID group configuration mode
  cns-connect-intf-config   CNS Connect Intf Info Mode
  config-l2tp-class         l2tp-class configuration mode
  config-rtr-http-rr        RTR HTTP raw request Configuration
  config-x25-huntgroup      X.25 hunt group configuration mode
  configure                 Global configuration mode
  congestion                Frame Relay congestion configuration mode
  controller                Controller configuration mode
  dhcp                      DHCP pool configuration mode
```

```
dnis-group              DNIS group configuration mode
exec                    Exec mode
filter                  Output filter mode
filterserver            AAA filter server definitions
flow-cache              Flow aggregation cache config mode
flow-sampler-map        Flow sampler map config mode
fr-fr                   FR/FR connection configuration mode
frf5                    FR/ATM Network IWF configuration mode
frf8                    FR/ATM Service IWF configuration mode
interface               Interface configuration mode
interface               Interface range configuration mode
interface-dlci          Frame Relay dlci configuration mode
ip-vrf                  Configure IP VRF parameters
ipenacl                 IP named extended access-list configuration mode
ipnat-pool              IP NAT pool configuration mode
ipnat-snat              IP SNAT configuration mode
ipnat-snat-backup       IP SNAT Backup configuration mode
ipnat-snat-primary      IP SNAT Primary configuration mode
ipnat-snat-redundancy   IP SNAT Redundancy configuration mode
ipsnacl                 IP named simple access-list configuration mode
iua-cfg                 ISDN user adaptation layer configuration
key-chain               Key-chain configuration mode
key-chain-key           Key-chain key configuration mode
kron-occurrence         Kron Occurrence SubMode
kron-policy             Kron Policy SubMode
line                    Line configuration mode
lw-vlan-id              VLAN-id configuration mode
lw-vlan-range           VLAN-range configuration mode
map-class               Map class configuration mode
map-list                Map list configuration mode
mrm-manager             IP Multicast Routing Monitor config mode
null-interface          Null interface configuration mode
policy-list             IP Policy List configuration mode
preauth                 AAA Preauth definitions
qosclassmap             QoS Class Map configuration mode
qosclasspolice          QoS Class Police configuration mode
qospolicymap            QoS Policy Map configuration mode
qospolicymapclass       QoS Policy Map class configuration mode
radius-attrl            Radius Attribute-List Definition
red-group               random-detect group configuration mode
request-dialin          VPDN group request dialin configuration mode
request-dialout         VPDN group request dialout configuration mode
roles                   Role configuration mode
route-map               Route map config mode
router                  Router configuration mode
rsvp-local-policy       RSVP local policy configuration mode
rtr                     SAA entry configuration
saa-dhcp                SAA dhcp configuration
saa-dns                 SAA dns configuration
saa-echo                SAA echo configuration
saa-frameRelay          SAA FrameRelay configuration
saa-ftp                 SAA ftp configuration
saa-http                SAA http configuration
saa-jitter              SAA jitter configuration
saa-pathEcho            SAA pathEcho configuration
saa-pathJitter          SAA pathJitter configuration
saa-slm-ctrlr-if        SAA SLM controller/interface configuration
saa-slmFrIf             SAA SLM FrameRelay Interface configuration
saa-slmfr               SAA SLM Frame Relay configuration
saa-tcpConnect          SAA tcpConnect configuration
saa-udpEcho             SAA udpEcho configuration
sg-radius               Radius Server-group Definition
sg-tacacs+              Tacacs+ Server-group Definition
signaling-class         Signaling class configuration mode
```

```
  sss-subscriber           SSS subscriber configuration mode
  subinterface             Subinterface configuration mode
  subscriber-policy        Subscriber policy configuration mode
  tablemap                 Table Map configuration mode
  tdm-conn                 TDM connection configuration mode
  template                 Template configuration mode
  tracking-config          Tracking configuration mode
  trange                   time-range configuration mode
  trunk-group              Trunk group configuration mode
  vc-class                 VC class configuration mode
  vc-group                 VC group configuration mode
  vlan                     VLAN database editing buffer
  vpdn-group               VPDN group configuration mode
  vpdn-template            VPDN template configuration mode
  x25-profile              X.25 profile configuration mode
```

In the following example, only commands in RTR Configuration mode are shown:

```
Router# show parser dump rtr
 Mode Name :rtr
15 type udpEcho dest-ipaddr <address> dest-port <1-65535> source-ipaddr <address>
source-port <1-65535> control enable
15 type udpEcho dest-ipaddr <address> dest-port <1-65535> source-ipaddr <address>
source-port <1-65535> control disable
15 type udpEcho dest-ipaddr <address> dest-port <1-65535> source-ipaddr <address>
source-port <1-65535>
15 type udpEcho dest-ipaddr <address> dest-port <1-65535> source-ipaddr <address>
15 type udpEcho dest-ipaddr <address> dest-port <1-65535>
15 type tcpConnect dest-ipaddr <address> dest-port <1-65535> source-ipaddr <address>
source-port <1-65535> control enable
15 type tcpConnect dest-ipaddr <address> dest-port <1-65535> source-ipaddr <address>
source-port <1-65535> control disable
15 type tcpConnect dest-ipaddr <address> dest-port <1-65535> source-ipaddr <address>
source-port <1-65535>
15 type tcpConnect dest-ipaddr <address> dest-port <1-65535> source-ipaddr <address>
15 type tcpConnect dest-ipaddr <address> dest-port <1-65535>
15 type jitter dest-ipaddr <address> dest-port <1-65535> source-ipaddr <address>
15 type jitter dest-ipaddr <address> dest-port <1-65535> source-port <1-65535>
15 type jitter dest-ipaddr <address> dest-port <1-65535> control enable
15 type jitter dest-ipaddr <address> dest-port <1-65535> control disable
15 type jitter dest-ipaddr <address> dest-port <1-65535> num-packets <1-60000>
15 type jitter dest-ipaddr <address> dest-port <1-65535> interval <1-60000>
15 type jitter dest-ipaddr <address> dest-port <1-65535>
15 type echo protocol ipIcmpEcho <address> source-ipaddr <address>
15 type echo protocol ipIcmpEcho <address>
15 type ftp operation get url <string> source-ipaddr <address> mode active
15 type ftp operation get url <string> source-ipaddr <address> mode passive
15 type ftp operation get url <string> source-ipaddr <address>
15 type ftp operation get url <string>
15 type http operation get url <string> name-server <address> version <string>
source-ipaddr <address> source-port <1-65535> cache
15 type http operation get url <string> name-server <address> version <string>
source-ipaddr <address> source-port <1-65535> cache
15 type http operation get url <string> name-server <address> version <string>
source-ipaddr <address> source-port <1-65535> cache
15 type http operation get url <string> name-server <address> version <string>
source-ipaddr <address> source-port <1-65535>
15 type http operation get url <string> name-server <address> version <string>
source-ipaddr <address>
15 type http operation get url <string> name-server <address> version <string>
15 type http operation get url <string> name-server <address>
15 type http operation get url <string>
15 type http operation raw
```

```
15 type dhcp dest-ipaddr <address> source-ipaddr <address> option <82-82> circuit-id
<string>
15 type dhcp dest-ipaddr <address> source-ipaddr <address> option <82-82> remote-id
<string>
15 type dhcp dest-ipaddr <address> source-ipaddr <address> option <82-82> subnet-mask
<ipmask>
15 type dhcp dest-ipaddr <address> source-ipaddr <address> option <82-82>
15 type dhcp dest-ipaddr <address> source-ipaddr <address>
15 type dhcp dest-ipaddr <address>
15 type dhcp
15 type dns target-addr <string> name-server <address> source-ipaddr <address> source-port
<1-65535>
15 type dns target-addr <string> name-server <address> source-ipaddr <address>
15 type dns target-addr <string> name-server <address>
15 type pathEcho protocol ipIcmpEcho <address> source-ipaddr <address>
15 type pathEcho protocol ipIcmpEcho <address>
15 type pathJitter dest-ipaddr <address> source-ipaddr <address>
15 type pathJitter dest-ipaddr <address> num-packets <1-100>
15 type pathJitter dest-ipaddr <address> interval <1-1000>
15 type pathJitter dest-ipaddr <address> targetOnly
15 type pathJitter dest-ipaddr <address>
15 type slm frame-relay pvc
15 type slm controller T1 <controller>
15 type slm controller E1 <controller>
15 type slm controller T3 <controller>
15 type slm controller E3 <controller>
15 exit
```

In the following example, only those commands in RTR Configuration mode containing the keyword **dhcp** are shown:

```
Router# show parser dump rtr | include dhcp
15 type dhcp dest-ipaddr <address> source-ipaddr <address> option <82-82> circuit-id
<string>
15 type dhcp dest-ipaddr <address> source-ipaddr <address> option <82-82> remote-id
<string>
15 type dhcp dest-ipaddr <address> source-ipaddr <address> option <82-82> subnet-mask
<ipmask>
15 type dhcp dest-ipaddr <address> source-ipaddr <address> option <82-82>
15 type dhcp dest-ipaddr <address> source-ipaddr <address>
15 type dhcp dest-ipaddr <address>
15 type dhcp
Router#
```

The following example shows how the **extend** keyword displays the syntax descriptions that match those shown using the ? command-line help:

```
Router# show parser dump rtr extend

 Mode Name :rtr
15 type udpEcho dest-ipaddr <address> dest-port <1-65535> source-ipaddr <address>
source-port <1-65535> control enable
type : Type of entry
udpEcho : UDP Echo Operation
dest-ipaddr : Destination address
<address> : IP address or hostname
dest-port : Destination Port
<1-65535> : Port Number
source-ipaddr : Source address
<address> : IP address or hostname
source-port : Source Port
<1-65535> : Port Number
control : Enable or disable control packets
```

```
enable : Enable control packets exchange (default)
 .
 .
 .
! Ctrl-Z used here to interrupt output and return to CLI prompt.

Router# config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# rtr 1
Router(config-rtr)# type udpEcho ?
  dest-ipaddr  Destination address

Router(config-rtr)# type udpEcho dest-ipaddr ?
  Hostname or A.B.C.D  IP address or hostname

Router(config-rtr)# type udpEcho dest-ipaddr HOSTNAME ?
  dest-port  Destination Port

Router(config-rtr)# type udpEcho dest-ipaddr HOSTNAME dest-port ?
  <1-65535>  Port Number

Router(config-rtr)# type udpEcho dest-ipaddr HOSTNAME dest-port 1 ?
  control        Enable or disable control packets
  source-ipaddr  Source address
  source-port    Source Port
  <cr>

Router(config-rtr)# type udpEcho dest-ipaddr HOSTNAME dest-port 1 control ?
  disable  Disable control packets exchange
  enable   Enable control packets exchange (default)
```

In the following example, show parser dump output is redirected to a file on a remote TFTP server:

```
show parser dump exec extend | redirect
tftp://209.165.200.225/userdirectory/123-exec-commands.txt
```

Related Commands

Command	Description
show <command> append	Redirects and adds the output of any **show** command to an existing file.
show <command> redirect	Redirects the output of any **show** command to a file.
show <command> tee	Copies the output of any **show** command to a file while displaying it on the terminal.
show <command> include	Filters **show** command output so that only lines that containing the specified string are displayed.
show <command> begin	Filters the output of any **show** command to display the output from the first instance of a specified string.
show <command> exclude	Filters **show** command output so that it excludes lines that contain a particular regular expression.

show parser macro

To display the smart port macros, use the **show parser macro** command in privileged EXEC mode.

show parser macro [**name** *macro-name* | **brief** | **description** [**interface** *interface*]]

Syntax Description

name *macro-name*	(Optional) Displays a specific macro.
brief	(Optional) Displays the configured macro names.
description	(Optional) Displays the macro description for all interfaces.
interface *interface*	(Optional) Displays the macro description for the specified interface.

Defaults

This command has no default settings.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.2(33)SXH	This command was introduced.

Examples

The following example shows how to display the macro description:

```
Router# show parser macro description

Interface    Macro Description

---------------------------------------------------------------

Fa1/2      desktop-config

---------------------------------------------------------------
```

The following example shows how to display the contents of the cisco-router smart port macro:

```
Router# show parser macro name cisco-router

Macro name : cisco-router
Macro type : default interface
# macro keywords $NVID
# Do not apply to EtherChannel/Port Group
# Access Uplink to Distribution
 switchport
# Define unique Native VLAN on trunk ports
# Recommended value for native vlan (NVID) should not be 1
 switchport trunk native vlan $NVID
# Update the allowed VLAN range (VRANGE) such that it
# includes data, voice and native VLANs
# switchport trunk allowed vlan VRANGE
# Hardcode trunk and disable negotiation to
# speed up convergence
 switchport trunk encapsulation dot1q
```

```
 switchport mode trunk
 switchport nonegotiate
# Configure qos to trust this interface
 auto qos voip trust
 mls qos trust dscp
# Ensure fast access to the network when enabling the interface.
# Ensure that switch devices cannot become active on the interface.
 spanning-tree portfast
 spanning-tree bpduguard enable
```

The following example shows how to list the Cisco-provided smart port macros:

```
Router# show parser macro brief | include default

    default global   :  cisco-global
    default interface:  cisco-desktop
    default interface:  cisco-phone
    default interface:  cisco-switch
    default interface:  cisco-router
```

Related Commands

Command	Description
macro (global configuration)	Creates a command macro.
macro (interface configuration)	Creates an interface-specific command macro.

show parser statistics

To displays statistics about the last configuration file parsed and the status of the Parser Cache feature, use the **show parser statistics** command in privileged EXEC mode.

show parser statistics

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(5)T	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

The **show parser statistics** command displays two sets of data:

- The number of commands in the configuration file that was last copied into the running configuration, and the time it took for the system to parse them (a configuration file can be loaded into the running configuration at system startup, or by issuing commands such as the **copy** *source* **running-config** command).
- The status of the Parser Cache feature (enabled or disabled) and the number of command matches (indicated by hits/misses) since the system was started or since the parser cache was cleared.

The Parser Cache feature optimizes the parsing (translation and execution) of Cisco IOS software configuration command lines by remembering how to parse recently encountered command lines, decreasing the time required to process large configuration files.

Examples

The following example shows sample output from the **show parser statistics** command:

```
Router# show parser statistics

Last configuration file parsed:Number of Commands:1484, Time:1272 ms

Parser cache:disabled, 0 hits, 2 misses
```

In this example, the Parser Cache feature is disabled, but shows the hit/miss statistics for the two commands issued while the parser cache was last enabled.

Table 118 describes the key output fields.

Table 118 show parser statistics Output Fields

Last configuration file parsed:	Displays statistics on the last configuration file copied into the running configuration (at startup or using the **copy** command).
Number of commands:	The number of command lines in the last configuration file parsed.
Time:	Time (in milliseconds) taken for the system to load the last configuration file.
Parser cache:	Displays whether the Parser Cache feature is enabled or disabled, and the hit/miss statistics related to the feature. Statistics are stored since the initialization of the system, or since the last time the parser cache was cleared.
hits	Number of commands the parser cache was able to parse more efficiently by matching them to similar commands executed previously.
misses	Number of commands the parser cache was unable to match to previously executed commands. The performance enhancement provided by the Parser Cache feature cannot be applied to unmatched commands.

In the following example the **show parser statistics** command is used to compare the parse-time of a large configuration file with the Parser Cache feature disabled and enabled. In this example, a configuration file with 1484 access list commands is loaded into the running configuration.

```
Router# configure terminal
!parser cache is disabled
Router(config)# no parser cache
!configuration file is loaded into the running configuration
Router# copy slot0:acl_list running-config
.
.
.
Router# show parser statistics
Last configuration file parsed:Number of Commands:1484, Time:1272 ms

Parser cache:disabled, 0 hits, 2 misses

!the parser cache is reenabled
Router(config)# parser cache
!configuration file is loaded into the running configuration
Router# copy slot0:acl_list running-config
.
.
.
Router# show parser statistics
Last configuration file parsed:Number of Commands:1484, Time:820 ms

Parser cache:enabled, 1460 hits, 26 misses
```

These results show an improvement to the load time for the same configuration file from 1272 milliseconds (ms) to 820 ms when the Parser Cache feature was enabled. As indicated in the "hits" field of the **show** command output, 1460 commands were able to be parsed more efficiently by the parser cache.

Related Commands

Command	Description
clear parser cache	Clears the parse cache entries and hit/miss statistics stored for the Parser Cache feature.
parser cache	Enables or disables the Parser Cache feature.

show pci

To display information about the peripheral component interconnect (PCI) hardware registers or bridge registers for the Cisco 7200 series routers, use the **show pci** command in EXEC mode.

show pci {**hardware** | **bridge** [*register*]}

Syntax Description

hardware	Displays PCI hardware registers.
bridge	Displays PCI bridge registers.
register	(Optional) Number of a specific bridge register in the range from 0 to 7. If not specified, this command displays information about all registers.

Command Modes

EXEC

Command History

Release	Modification
11.2	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

The output of this command is generally useful for diagnostic tasks performed by technical support only.

Note The **show pci hardware** EXEC command displays a substantial amount of information.

Examples

The following is sample output for the PCI bridge register 1 on a Cisco 7200 series router:

```
Router# show pci bridge 1

Bridge 4, Port Adaptor 1, Handle=1
DEC21050 bridge chip, config=0x0
(0x00): cfid   = 0x00011011
(0x04): cfcs   = 0x02800147
(0x08): cfccid = 0x06040002
(0x0C): cfpmlt = 0x00010010

(0x18): cfsmlt = 0x18050504
(0x1C): cfsis  = 0x22805050
(0x20): cfmla  = 0x48F04880
(0x24): cfpmla = 0x00004880

(0x3C): cfbc   = 0x00000000
(0x40): cfseed = 0x00100000
(0x44): cfstwt = 0x00008020
```

The following is partial sample output for the PCI hardware register, which also includes information on all the PCI bridge registers on a Cisco 7200 series router:

```
Router# show pci hardware
```

```
GT64010 External PCI Configuration registers:
 Vendor / Device ID   : 0xAB114601 (b/s 0x014611AB)
 Status / Command     : 0x17018002 (b/s 0x02800117)
 Class / Revision     : 0x00000006 (b/s 0x06000000)
 Latency              : 0x0F000000 (b/s 0x0000000F)
 RAS[1:0] Base        : 0x00000000 (b/s 0x00000000)
 RAS[3:2] Base        : 0x00000001 (b/s 0x01000000)
 CS[2:0] Base         : 0x00000000 (b/s 0x00000000)
 CS[3] Base           : 0x00000000 (b/s 0x00000000)
 Mem Map Base         : 0x00000014 (b/s 0x14000000)
 IO Map Base          : 0x01000014 (b/s 0x14000001)
 Int Pin / Line       : 0x00010000 (b/s 0x00000100)

Bridge 0, Downstream MB0 to MB1, Handle=0
DEC21050 bridge chip, config=0x0
(0x00): cfid   = 0x00011011
(0x04): cfcs   = 0x02800143
(0x08): cfccid = 0x06040002
(0x0C): cfpmlt = 0x00011810

(0x18): cfsmlt = 0x18000100
(0x1C): cfsis  = 0x02809050
(0x20): cfmla  = 0x4AF04880
(0x24): cfpmla = 0x4BF04B00

(0x3C): cfbc   = 0x00000000
(0x40): cfseed = 0x00100000
(0x44): cfstwt = 0x00008020
.
.
.
```

show pci hardware

To display information about the Host-PCI bridge, use the **show pci hardware** command in EXEC mode.

show pci hardware

Syntax Description

This command has no arguments or keywords.

Command Modes

EXEC

Command History

Release	Modification
11.2	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

The output of this command is generally useful for diagnostic tasks performed by technical support only:

```
Router# show pci hardware

hardware PCI hardware registers

Each device on the PCI bus is assigned a PCI device number.  For the
C2600, device numbers are as follows:

Device    Device number
0         First LAN device
1         Second LAN device
2         AIM device (if present)
3         Not presently used
4         Port module - first PCI device
5         Port module - second PCI device
6         Port module - third PCI device
7         Port module - fourth PCI device
8-14      Not presently used
15        Xilinx PCI bridge
```

Examples

The following is partial sample output for the PCI hardware register, which also includes information on all the PCI bridge registers.

```
router# show pci hardware

XILINX Host-PCI Bridge Registers:
Vendor / Device ID: 0x401310EE
Status / Command: 0x040001C6
PCI Slave Base Reg 0: 0x00000000
PCI Slave Base Reg 1: 0x04000000
```

Table 119 describes the significant fields shown in the display.

Table 119 show pci hardware Field Descriptions

Field	Description
Device/Vendor ID	Identifies the PCI vendor and device. The value 0x401310EE identifies the device as the Xilinx-based Host-PCI bridge for the Cisco 2600 router.
Status/Command	Provides status of the Host-PCI bridge. Refer to the PCI Specification for more information.
PCI Slave Base Reg 0	The base address of PCI Target Region 0 for the Host-PCI bridge. This region is used for Big-Endian transfers between PCI devices and memory.
PCI Slave Base Reg 1	The base address of PCI Target Region 1 for the Host-PCI bridge. This region is used for Little-Endian transfers between PCI devices and memory.

show platform

To display platform information, use the **show platform** command privileged EXEC mode.

show platform {**buffers** | **copp rate-limit** {**arp** | **dhcp** | **atm-oam** | **ethernet-oam** | **pppoe-discovery** | **all**} | **np copp** [*ifnum*] [**detail**] | **eeprom** | **fault** | **hardware capacity** | **hardware pfc mode** | **internal-vlan** | **netint** | **software ipv6-multicast connected** | **tech-support ipmulticast** *group-ip-addr src-ip-addr* | **tlb**}

Syntax Description

buffers	Displays buffer-allocation information.
copp rate-limit	Displays CoPP rate-limit information on the Cisco 7600 SIP-400.
arp	Specifies ARP protocol packet traffic.
dhcp	Specifies DHCP protocol packet traffic.
atm-oam	Specifies ATM OAM packet traffic.
ethernet-oam	Specifies Ethernet OAM packet traffic.
pppoe-discovery	Specifies PPPoE discovery packet information.
all	Displays rate-limit information for all protocols.
np copp	Displays debug information for a given CoPP session ID or for all CoPP sessions.
ifnum	Specifies a session ID.
detail	Shows full rate-limited values.
eeprom	Displays CPU EEPROM information.
fault	Displays the fault date.
hardware capacity	Displays the capacities and utilizations for hardware resources; see the show platform hardware capacity command.
hardware pfc mode	Displays the type of installed PFC.
internal-vlan	Displays the internal VLAN.
netint	Displays the platform network-interrupt information.
software ipv6-multicast connected	Displays all the IPv6 subnet ACL entries on the route processor; see the show platform software ipv6-multicast command.
tech-support ipmulticast	Displays IP multicast-related information for TAC.
group-ip-addr	Group IP address.
src-ip-addr	Source IP address.
tlb	Displays information about the TLB register.

Defaults

This command has no default settings.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.2(14)SX	Support for this command was introduced on the Supervisor Engine 720.
12.2(17d)SXB	Support for this command on the Supervisor Engine 2 was extended to Release 12.2(17d)SXB. This command was changed to include the **hardware pfc mode** keywords.
12.2(18)SXD	This command was changed to include the **software ipv6-multicast connected** keywords.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SRC	This command was modified to include additional keywords to support CoPP enhancements on the Cisco SIP-400 on the Cisco 7600 series router.

Usage Guidelines

This command is similar to the **show msfc** command.

Examples

This example shows how to display buffer-allocation information:

```
Router# show platform buffers

Reg. set    Min    Max
  TX               640
 ABQ         640  16384
   0           0     40
   1        6715   8192
   2           0      0
   3           0      0
   4           0      0
   5           0      0
   6           0      0
   7           0      0
Threshold = 8192

Vlan  Sel  Min  Max  Cnt  Rsvd
1019    1 6715 8192    0     0
Router#
```

Cisco 7600 Series Routers with Cisco 7600 SIP-400

This example shows how to display the list of interfaces on which a rate limiter is active for Address Resolution Protocol (ARP), along with the count of confirmed and exceeded packets for the rate limiter.

```
Router# show platform copp rate-limit arp
Rate limiter Information for Protocol arp:

  Rate Limiter Status: Enabled
  Rate : 20 pps
  Max Observation Period : 60 seconds
Per Interface Rate Limiter Information
  Interface              Conformed Pkts  Exceeded Pkts  Enabled  Obs Period (Mts)
  GigabitEthernet5/1         0               0          No        -
  GigabitEhternet5/1.1       14              0          No        -
  GigabitEthernet5/1.2       28              2          No        -
  GigabitEthernet5/2         0               0          No        -
  GigabitEthernet5/2.1       180             4          Yes       35
  GigabitEthernet5/2.2       200             16         Yes       Max
Router#
```

Table 120 describes the significant fields shown in the display.

Table 120 show platform copp rate-limit Field Descriptions

Field	Description
Rate Limiter Status	Indicates if a rate limiter has been enabled on the interface.
Rate	Indicates the configured rate in pps or bps.
Max Observation Period	Indicates the configured observation period before automatically turning off the per-interface rate limiter.
Per Interface Rate Limiter Information	Displays the list of interfaces on which the rate limiter is active. In this example: • GigabitEthernet5/1.1 is free from attack. • GigabitEthernet5/2.1 has an exceed count of 4, and has a rate limiter enabled. The observation period is 35 minutes, which indicates that currently the interface is free from attack and is being kept under observation. The interface will remain under observation for an additional 35 minutes. If it remains free from attack after that time, the rate limiter is automatically removed. • GigabitEthernet5/2.2 has an exceed count of 16 and has a rate limiter enabled. The observation period has been designated as Max. This indicates that the interface is still under attack and has not yet entered the observation time window.

This example shows how to display CPU EEPROM information:

```
Router# show platform eeprom

MSFC CPU IDPROM:
IDPROM image:

IDPROM image block #0:
  hexadecimal contents of block:
  00: AB AB 02 9C 13 5B 02 00 00 02 60 03 03 E9 43 69    .....[....`...Ci
  10: 73 63 6F 20 53 79 73 74 65 6D 73 00 00 00 00 00    sco Systems.....
  20: 00 00 57 53 2D 58 36 4B 2D 53 55 50 33 2D 50 46    ..WS-X6K-SUP3-PF
  30: 43 33 00 00 00 00 53 41 44 30 36 34 34 30 31 57    C3....SAD064401W
  40: 4C 00 00 00 00 00 00 00 00 00 37 33 2D 37 34 30    L.........73-740
  50: 34 2D 30 37 00 00 00 00 00 00 30 35 00 00 00 00    4-07......05....
  60: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
  70: 00 00 00 00 02 BD 00 00 00 00 00 09 00 05 00 01    ................
  80: 00 03 00 01 00 01 00 02 03 E9 00 00 00 00 00 00    ................
  90: 00 00 00 00 00 00 00 00 00 00 00 00                ............

  block-signature = 0xABAB, block-version = 2,
  block-length = 156, block-checksum = 4955

  *** common-block ***
  IDPROM capacity (bytes) = 512  IDPROM block-count = 2
  FRU type = (0x6003,1001)
  OEM String = 'Cisco Systems'
  Product Number = 'WS-X6K-SUP3-PFC3'
  Serial Number = 'SAD064401WL'
  Manufacturing Assembly Number = '73-7404-07'
  Manufacturing Assembly Revision = '05'
```

```
  Hardware Revision = 0.701
  Manufacturing bits = 0x0  Engineering bits = 0x0
  SNMP OID = 9.5.1.3.1.1.2.1001
  Power Consumption = 0 centiamperes    RMA failure code = 0-0-0-0
  CLEI =
  *** end of common block ***

IDPROM image block #1:
  hexadecimal contents of block:
  00: 60 03 02 67 0C 24 00 00 00 00 00 00 00 00 00 00    `..g.$..........
  10: 00 00 00 00 00 00 00 51 00 05 9A 3A 7E 9C 00 00    .......Q...:~...
  20: 02 02 00 01 00 01 00 00 00 00 00 00 00 00 00 00    ................
  30: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
  40: 14 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
  50: 00 00 81 81 81 81 80 80 80 80 80 80 80 80 80 80    ................
  60: 80 80 06 72 00 46 37                               ...r.F7

  block-signature = 0x6003, block-version = 2,
  block-length = 103, block-checksum = 3108

  *** linecard specific block ***
  feature-bits =   00000000 00000000
  hardware-changes-bits =   00000000 00000000
  card index = 81
  mac base = 0005.9A3A.7E9C
  mac_len = 0
  num_processors = 2
  epld_num = 2
  epld_versions = 0001 0001 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000
  port numbers:
    pair #0: type=14, count=01
    pair #1: type=00, count=00
    pair #2: type=00, count=00
    pair #3: type=00, count=00
    pair #4: type=00, count=00
    pair #5: type=00, count=00
    pair #6: type=00, count=00
    pair #7: type=00, count=00
  sram_size = 0
  sensor_thresholds =
    sensor #0: critical = -127 oC (sensor present but ignored), warning = -127 oC (sensor
present but ignored)
    sensor #1: critical = -127 oC (sensor present but ignored), warning = -127 oC (sensor
present but ignored)
    sensor #2: critical = -128 oC (sensor not present), warning = -128 oC (sensor not
present)
    sensor #3: critical = -128 oC (sensor not present), warning = -128 oC (sensor not
present)
    sensor #4: critical = -128 oC (sensor not present), warning = -128 oC (sensor not
present)
    sensor #5: critical = -128 oC (sensor not present), warning = -128 oC (sensor not
present)
    sensor #6: critical = -128 oC (sensor not present), warning = -128 oC (sensor not
present)
    sensor #7: critical = -128 oC (sensor not present), warning = -128 oC (sensor not
present)
  max_connector_power = 1650
  cooling_requirement = 70
  ambient_temp = 55
  *** end of linecard specific block ***
Router#
```

This example shows how to display fault-date information:

```
Router# show platform fault

Fault History Buffer:
s72033_rp Software (s72033_rp-JSV-M), Experimental Version 12.2(20030331:071521)
[kkuttuva-CSCea55513-const2 120]
Compiled Mon 31-Mar-03 21:58 by kkuttuva
Signal = 10, Code = 0x1C, Uptime 00:01:14
$0 : 00000000, AT : 00000000, v0 : 00000000, v1 : 00000000
a0 : 00000000, a1 : 10050000, a2 : 00000000, a3 : 43F4B614
t0 : 50A19548, t1 : 10048000, t2 : 10040000, t3 : 10050000
t4 : 43F515A8, t5 : 43F515A4, t6 : 43F515A0, t7 : 43F5159C
s0 : 50A19548, s1 : 00000000, s2 : 50A19548, s3 : 10030100
s4 : 10030000, s5 : 41700000, s6 : 43F4B614, s7 : 41DB0000
t8 : 43F51614, t9 : 00000000, k0 : 5032D19C, k1 : 40231598
gp : 41F96960, sp : 50A19508, s8 : 422183A0, ra : 4027FB50
EPC : 4027FB84, SREG : 3401F103, Cause : 8000001C
Router#
```

This example shows how to display the PFC-operating mode:

```
Router# show platform hardware pfc mode

PFC operating mode : PFC3A
Router#
```

This example shows how to display platform net-interrupt information:

```
Router# show platform netint

Network IO Interrupt Throttling:
 throttle count=0, timer count=0
 active=0, configured=1
 netint usec=3999, netint mask usec=800
inband_throttle_mask_hi = 0x0
inband_throttle_mask_lo = 0x800000
Router#
```

This example shows how to display TLB-register information:

```
Router# show platform tlb

Mistral revision 5
TLB entries : 42
Virt Address range      Phy Address range      Attributes
0x10000000:0x1001FFFF   0x010000000:0x01001FFFF   CacheMode=2, RW, Valid
0x10020000:0x1003FFFF   0x010020000:0x01003FFFF   CacheMode=2, RW, Valid
0x10040000:0x1005FFFF   0x010040000:0x01005FFFF   CacheMode=2, RW, Valid
0x10060000:0x1007FFFF   0x010060000:0x01007FFFF   CacheMode=2, RW, Valid
0x10080000:0x10087FFF   0x010080000:0x010087FFF   CacheMode=2, RW, Valid
0x10088000:0x1008FFFF   0x010088000:0x01008FFFF   CacheMode=2, RW, Valid
0x18000000:0x1801FFFF   0x010000000:0x01001FFFF   CacheMode=0, RW, Valid
0x19000000:0x1901FFFF   0x010000000:0x01001FFFF   CacheMode=7, RW, Valid
0x1E000000:0x1E1FFFFF   0x01E000000:0x01E1FFFFF   CacheMode=2, RW, Valid
0x1E880000:0x1E899FFF   0x01E880000:0x01E899FFF   CacheMode=2, RW, Valid
0x1FC00000:0x1FC7FFFF   0x01FC00000:0x01FC7FFFF   CacheMode=2, RO, Valid
0x30000000:0x3001FFFF   0x070000000:0x07001FFFF   CacheMode=2, RW, Valid
0x40000000:0x407FFFFF   0x000000000:0x0007FFFFF   CacheMode=3, RO, Valid
.
```

```
.
.
0x58000000:0x59FFFFFF   0x088000000:0x089FFFFFF   CacheMode=3, RW, Valid
0x5A000000:0x5BFFFFFF   0x08A000000:0x08BFFFFFF   CacheMode=3, RW, Valid
0x5C000000:0x5DFFFFFF   0x08C000000:0x08DFFFFFF   CacheMode=3, RW, Valid
0x5E000000:0x5FFFFFFF   0x08E000000:0x08FFFFFFF   CacheMode=3, RW, Valid
Router#
```

Related Commands

Command	Description
show msfc	Displays MSFC information.
Cisco 7600 series router with Cisco 7600 SIP-400	
platform copp	Turns on or off rate-limiting for an interface on the Cisco 7600 SIP-400.
platform copp observation period	Sets the observation period before automatically turning off the per-interface rate limiter on the Cisco 7600 SIP-400.

show platform hardware capacity

To display the capacities and utilizations for the hardware resources, use the **show platform hardware capacity** command in privileged EXEC mode.

show platform hardware capacity [*resource-type*]

Syntax Description		
	resource-type	(Optional) Hardware resource type; see the "Usage Guidelines" section for the valid values.

Defaults

This command has no default settings.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(18)SXF	Support for this command was introduced on the Supervisor Engine 720 and the Supervisor Engine 2.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

The valid values for *resource-type* are as follows:

- **acl**—Displays the capacities and utilizations for ACL/QoS TCAM resources.
- **cpu**—Displays the capacities and utilizations for CPU resources.
- **eobc**—Displays the capacities and utilizations for EOBC resources.
- **fabric**—Displays the capacities and utilizations for Switch Fabric resources.
- **flash**—Displays the capacities and utilizations for Flash/NVRAM resources.
- **forwarding**—Displays the capacities and utilizations for Layer 2 and Layer 3 forwarding resources.
- **interface**—Displays the capacities and utilizations for interface resources.
- **monitor**—Displays the capacities and utilizations for SPAN resources.
- **multicast**—Displays the capacities and utilizations for Layer 3 multicast resources.
- **netflow**—Displays the capacities and utilizations for NetFlow resources.
- **pfc**—Displays the capacities and utilizations for all the PFC resources including Layer 2 and Layer 3 forwarding, NetFlow, CPU rate limiters, and ACL/QoS TCAM resources.
- **power**—Displays the capacities and utilizations for power resources.
- **qos**—Displays the capacities and utilizations for QoS policer resources.
- **rate-limiter**—Displays the capacities and utilizations for CPU rate limiter resources.
- **system**—Displays the capacities and utilizations for system resources.
- **vlan**—Displays the capacities and utilizations for VLAN resources.

The **show platform hardware capacity cpu** command displays the following information:

- CPU utilization for the last 5 seconds (busy time and interrupt time), the percentage of the last 1-minute average busy time, and the percentage of the last 5-minute average busy time.
- Processor memory total available bytes, used bytes, and percentage used.
- I/O memory total available bytes, used bytes, and percentage used.

The **show platform hardware capacity eobc** command displays the following information:

- Transmit and receive rate
- Packets received and packets sent
- Dropped received packets and dropped transmitted packets

The **show platform hardware capacity forwarding** command displays the following information:

- The total available entries, used entries, and used percentage for the MAC tables.
- The total available entries, used entries, and used percentage for the FIB TCAM tables. The display is done per protocol base.
- The total available entries, used entries, and used percentage for the adjacency tables. The display is done for each region in which the adjacency table is divided.
- The created entries, failures, and resource usage percentage for the NetFlow TCAM and ICAM tables.
- The total available entries and mask, used entries and mask, reserved entries and mask, and entries and mask used percentage for the ACL/QoS TCAM tables. The output displays the available, used, reserved, and used percentage of the labels. The output displays the resource of other hardware resources that are related to the ACL/QoS TCAMs (such as available, used, reserved, and used percentage of the LOU, ANDOR, and ORAND).
- The available, used, reserved, and used percentage for the CPU rate limiters.

The **show platform hardware capacity interface** command displays the following information:

- Tx/Rx drops—Displays the sum of transmit and receive drop counters on each online module (aggregate for all ports) and provides the port number that has the highest drop count on the module.
- Tx/Rx per port buffer size—Summarizes the port-buffer size on a per-module basis for modules where there is a consistent buffer size across the module.

The **show platform hardware capacity monitor** command displays the following SPAN information:

- The maximum local SPAN sessions, maximum RSPAN sessions, maximum ERSPAN sessions, and maximum service module sessions.
- The local SPAN sessions used/available, RSPAN sessions used/available, ERSPAN sessions used/available, and service module sessions used/available.

The **show platform hardware capacity multicast** command displays the following information:

- Multicast Replication Mode: ingress and egress IPv4 and IPv6 modes.
- The MET table usage that indicates the total used and the percentage used for each module in the system.
- The bidirectional PIM DF table usage that indicates the total used and the percentage used.

The **show platform hardware capacity system** command displays the following information:

- PFC operating mode (PFC Version: PFC3A, PFC3B, unknown, and so forth)
- Supervisor redundancy mode (RPR, RPR+, SSO, none, and so forth)

- Module-specific switching information, including the following information:
  - Part number (WS-SUP720-BASE, WS-X6548-RJ-45, and so forth)
  - Series (supervisor engine, fabric, CEF720, CEF256, dCEF256, or classic)
  - CEF Mode (central CEF, dCEF)

The **show platform hardware capacity vlan** command displays the following VLAN information:

- Total VLANs
- VTP VLANs that are used
- External VLANs that are used
- Internal VLANs that are used
- Free VLANs

Examples

This example shows how to display CPU capacity and utilization information for the route processor, the switch processor, and the LAN module in the Cisco 7600 series router:

```
Router# show platform hardware capacity cpu

CPU Resources
  CPU utilization: Module             5 seconds       1 minute       5 minutes
                   1  RP               0% /  0%            1%             1%
                   1  SP               5% /  0%            5%             4%
                   7                  69% /  0%           69%            69%
                   8                  78% /  0%           74%            74%
  Processor memory: Module   Bytes:       Total          Used           %Used
                    1  RP              176730048       51774704           29%
                    1  SP              192825092       51978936           27%
                    7                  195111584       35769704           18%
                    8                  195111584       35798632           18%
  I/O memory: Module         Bytes:       Total          Used           %Used
              1  RP                     35651584       12226672           34%
              1  SP                     35651584        9747952           27%
              7                         35651584        9616816           27%
              8                         35651584        9616816           27%
Router#
```

This example shows how to display EOBC-related statistics for the route processor, the switch processor, and the DFCs in the Cisco 7600 series router:

```
Router# show platform hardware capacity eobc

EOBC Resources
  Module                     Packets/sec     Total packets     Dropped packets
  1  RP      Rx:                      61            108982                   0
             Tx:                      37             77298                   0
  1  SP      Rx:                      34            101627                   0
             Tx:                      39            115417                   0
  7          Rx:                       5             10358                   0
             Tx:                       8             18543                   0
  8          Rx:                       5             12130                   0
             Tx:                      10             20317                   0
Router#
```

This example shows how to display the current and peak switching utilization:

```
Router# show platform hardware capacity fabric

Switch Fabric Resources
```

```
  Bus utilization: current is 100%, peak was 100% at 12:34 12mar45
  Fabric utilization:      ingress                      egress
      Module channel speed current peak                 current peak
       1      0        20G   100%  100% 12:34 12mar45  100%    100% 12:34 12mar45

       1      1        20G   12%    80% 12:34 12mar45   12%     80% 12:34 12mar45
       4      0        20G   12%    80% 12:34 12mar45   12%     80% 12:34 12mar45
      13      0         8G   12%    80% 12:34 12mar45   12%     80% 12:34 12mar45
Router#
```

This example shows how to display information about the total capacity, the bytes used, and the percentage that is used for the Flash/NVRAM resources present in the system:

```
Router# show platform hardware capacity flash

Flash/NVRAM Resources
  Usage: Module Device               Bytes:      Total          Used     %Used
         1  RP  bootflash:                     31981568      15688048       49%
         1  SP  disk0:                        128577536     105621504       82%
         1  SP  sup-bootflash:                 31981568      29700644       93%
         1  SP  const_nvram:                     129004           856        1%
         1  SP  nvram:                           391160         22065        6%
         7      dfc#7-bootflash:               15204352        616540        4%
         8      dfc#8-bootflash:               15204352             0        0%
Router#
```

This example shows how to display the capacity and utilization of the EARLs present in the system:

```
Router# show platform hardware capacity forwarding

L2 Forwarding Resources
           MAC Table usage:   Module  Collisions  Total       Used       %Used
                              6                0  65536         11          1%

             VPN CAM usage:                        Total       Used       %Used
                                                     512          0          0%
L3 Forwarding Resources
              FIB TCAM usage:                     Total       Used       %Used
                   72 bits (IPv4, MPLS, EoM)     196608         36          1%
                  144 bits (IP mcast, IPv6)       32768          7          1%

                      detail:      Protocol                   Used       %Used
                                   IPv4                         36          1%
                                   MPLS                          0          0%
                                   EoM                           0          0%

                                   IPv6                          4          1%
                                   IPv4 mcast                    3          1%
                                   IPv6 mcast                    0          0%

            Adjacency usage:                     Total       Used       %Used
                                                1048576        175          1%

     Forwarding engine load:
                    Module       pps   peak-pps                    peak-time
                    6              8       1972  02:02:17 UTC Thu Apr 21 2005

Netflow Resources
          TCAM utilization:       Module       Created      Failed       %Used
                                  6                 1           0          0%
          ICAM utilization:       Module       Created      Failed       %Used
                                  6                 0           0          0%

                Flowmasks:   Mask#   Type        Features
```

```
                          IPv4:     0   reserved    none
                          IPv4:     1   Intf FulNAT_INGRESS NAT_EGRESS FM_GUARDIAN
                          IPv4:     2   unused      none
                          IPv4:     3   reserved    none

                          IPv6:     0   reserved    none
                          IPv6:     1   unused      none
                          IPv6:     2   unused      none
                          IPv6:     3   reserved    none

CPU Rate Limiters Resources
             Rate limiters:       Total         Used      Reserved       %Used
                    Layer 3           9            4             1         44%
                    Layer 2           4            2             2         50%
ACL/QoS TCAM Resources
  Key: ACLent - ACL TCAM entries, ACLmsk - ACL TCAM masks, AND - ANDOR,
       QoSent - QoS TCAM entries, QOSmsk - QoS TCAM masks, OR - ORAND,
       Lbl-in - ingress label, Lbl-eg - egress label, LOUsrc - LOU source,
       LOUdst - LOU destination, ADJ - ACL adjacency

  Module ACLent ACLmsk QoSent QoSmsk Lbl-in Lbl-eg LOUsrc LOUdst  AND  OR  ADJ
  6          1%     1%     1%     1%     1%     1%     0%     0%   0%  0%   1%

Router#
```

This example shows how to display the interface resources:

```
Router# show platform hardware capacity interface

Interface Resources
  Interface drops:
    Module    Total drops:    Tx            Rx      Highest drop port:  Tx  Rx
    9                          0             2                           0  48

  Interface buffer sizes:
    Module                             Bytes:     Tx buffer           Rx buffer
         1                                            12345               12345
         5                                            12345               12345
Router#
```

This example shows how to display SPAN information:

```
Router# show platform hardware capacity monitor

SPAN Resources
  Source sessions: 2 maximum, 0 used
    Type                                  Used
    Local                                    0
    RSPAN source                             0
    ERSPAN source                            0
    Service module                           0
  Destination sessions: 64 maximum, 0 used
    Type                                  Used
    RSPAN destination                        0
    ERSPAN destination (max  24)             0
Router#
```

This example shows how to display the capacity and utilization of resources for Layer 3 multicast functionality:

```
Router# show platform hardware capacity multicast

L3 Multicast Resources
  IPv4 replication mode: ingress
```

```
  IPv6 replication mode: ingress
  Bi-directional PIM Designated Forwarder Table usage: 4 total, 0 (0%) used
  Replication capability: Module                                IPv4        IPv6
                          5                                    egress      egress
                          9                                   ingress     ingress
  MET table Entries: Module                                 Total    Used    %Used
                     5                                      65526       6       0%
Router#
```

This example shows how to display information about the system power capacities and utilizations:

```
Router# show platform hardware capacity power

Power Resources
  Power supply redundancy mode: administratively combined
                                                      operationally combined
  System power: 1922W, 0W (0%) inline, 1289W (67%) total allocated
  Powered devices: 0 total
Router#
```

This example shows how to display the capacity and utilization of QoS policer resources per EARL in the Cisco 7600 series router:

```
Router# show platform hardware capacity qos

 QoS Policer Resources
  Aggregate policers: Module                      Total         Used     %Used
                      1                            1024          102       10%
                      5                            1024            1        1%
  Microflow policer configurations: Module        Total         Used     %Used
                                    1                64           32       50%
                                    5                64            1        1%
Router#
```

This example shows how to display information about the key system resources:

```
Router# show platform hardware capacity system

System Resources
  PFC operating mode: PFC3BXL
  Supervisor redundancy mode: administratively rpr-plus, operationally rpr-plus
  Switching Resources: Module   Part number              Series      CEF mode
                       5        WS-SUP720-BASE        supervisor           CEF
                       9        WS-X6548-RJ-45            CEF256           CEF
Router#
```

This example shows how to display VLAN information:

```
Router# show platform hardware capacity vlan

VLAN Resources
  VLANs: 4094 total, 10 VTP, 0 extended, 0 internal, 4084 free
Router#
```

Related Commands

Command	Description
show msfc	Displays MSFC information.
show platform	Displays platform information.

show platform software filesystem

To display information about file systems, use the **show platform software filesystem** command in privileged EXEC or diagnostic mode.

show platform software filesystem {**bootflash: | stby-bootflash: | fpd: | harddisk: | stby-harddisk: | obfl: | stby-obfl: | usb0: | stby-usb0: | usb1: | stby-usb1:**} [**all**] [**details**]

Syntax Description

bootflash:	File system on the bootflash device.
stby-bootflash:	Standby file system on the bootflash device (if the standby Route Processor [RP] is preset).
fpd:	Synthetic file system that is used by the field-programmable device (FPD) upgrade process—for Cisco Technical Support only.
harddisk:	File system on the hard disk device.
stby-harddisk:	Standby file system on the harddisk device (if the standby RP is preset).
obfl:	File system on the on board failure logging (OBFL) device.
stby-obfl:	Standby file system on the OBFL device (if the standby RP is preset).
usb0:	File system on the USB0 device (if installed).
stby-usb0:	Standby file system on the USB0 device (if the standby RP is preset).
usb1:	File system on the USB1 device (if installed).
stby-usb1:	Standby file system on the USB1 device (if the standby RP is preset).
all	(Optional) All possible device information.
details	(Optional) File system details.

Command Default

No default behavior or values

Command Modes

Privileged EXEC (#)

Diagnostic (diag)

Command History

Release	Modification
Cisco IOS XE Release 2.1	This command was introduced on the Cisco ASR1000 Series Routers.

Usage Guidelines

Use this command to ascertain the presence or absence of specific files and to determine space usage in the file system. This command is helpful to monitor the growth of log file sizes, because rapid growth of log files could indicate possible problems with the router.

Examples

The following example displays information about the files in the bootflash file system. It also shows the number of bytes used out of the total available in the bootflash file system.

```
Router# show platform software filesystem bootflash:

-#- --length-- ---------date/time--------- path
  1       4096 Apr 01 2008 13:34:30 +00:00 /bootflash/
  2      16384 Dec 04 2007 04:32:46 +00:00 /bootflash/lost+found
  3       4096 Dec 04 2007 06:06:24 +00:00 /bootflash/.ssh
  4        963 Dec 04 2007 06:06:16 +00:00 /bootflash/.ssh/ssh_host_key
  5        627 Dec 04 2007 06:06:16 +00:00 /bootflash/.ssh/ssh_host_key.pub
  6       1675 Dec 04 2007 06:06:18 +00:00 /bootflash/.ssh/ssh_host_rsa_key
  7        382 Dec 04 2007 06:06:18 +00:00 /bootflash/.ssh/ssh_host_rsa_key.pub
  8        668 Dec 04 2007 06:06:24 +00:00 /bootflash/.ssh/ssh_host_dsa_key
  9        590 Dec 04 2007 06:06:24 +00:00 /bootflash/.ssh/ssh_host_dsa_key.pub
 10       4096 Dec 04 2007 06:06:36 +00:00 /bootflash/.rollback_timer
 11       4096 Mar 18 2008 17:31:17 +00:00 /bootflash/.prst_sync
 12       4096 Dec 04 2007 04:34:45 +00:00 /bootflash/.installer
 13  205951180 Mar 18 2008 17:23:03 +00:00 /bootflash/asr1000rp1-advipservicesk
 14   46858444 Mar 18 2008 17:28:55 +00:00 /bootflash/asr1000rp1-espbase.02.01.
 15   20318412 Mar 18 2008 17:28:56 +00:00 /bootflash/asr1000rp1-rpaccess-k9.02
 16   22266060 Mar 18 2008 17:28:57 +00:00 /bootflash/asr1000rp1-rpbase.02.01.0
 17   21659852 Mar 18 2008 17:28:57 +00:00 /bootflash/asr1000rp1-rpcontrol.02.0
 18   45934796 Mar 18 2008 17:28:58 +00:00 /bootflash/asr1000rp1-rpios-advipser
 19   34169036 Mar 18 2008 17:28:59 +00:00 /bootflash/asr1000rp1-sipbase.02.01.
 20   22067404 Mar 18 2008 17:29:00 +00:00 /bootflash/asr1000rp1-sipspa.02.01.0
 21       7180 Mar 18 2008 17:29:00 +00:00 /bootflash/packages.conf

461897728 bytes available (419782656 bytes used)
```

The following example displays information only about the bootflash file system itself, such as file system type and access permissions:

```
Router# show platform software filesystem bootflash: details

Filesystem: bootflash
Filesystem Path: /bootflash
Filesystem Type: ext2
Mounted: Read/Write
```

Table 121 describes the significant fields shown in the displays of file system information.

Table 121 show platform software filesystem Field Descriptions

Field	Description
#	Display line number.
Length	File size in bytes.
Date/Time	Date and time the file system was created.
Path	Full path of a file in the file system.
Filesystem Path	Root of the file system.
Filesystem Type	Type of file system. One of the following values: • ext2—Second extended file system. • jffs2—Journaling flash file system, version 2. • vfat—Virtual file allocation table (FAT16 or FAT32).
Mounted	Access permissions to the file system.

Related Commands

Command	Description
show platform software mount	Displays the mounted file systems (both physical and virtual) on a shared port adapter (SPA) in a SPA interface processor (SIP), on an Embedded Services Processor (ESP), or on a Route Processor (RP).
show platform software tech-support	Displays system information or creates a technical support information tar file for Cisco Technical Support.

show platform software memory

To display memory information for the specified process, use the **show platform software memory** command in privileged EXEC or diagnostic mode.

show platform software memory [**database** | **messaging**] {**chassis-manager** *slot* | **cpp-control-process** *process* | **cpp-driver** *process* | **cpp-ha-server** *process* | **cpp-service-process** *process* | **forwarding-manager** *slot* | **host-manager** *slot* | **interface-manager** *slot* | **ios** *slot* | **logger** *slot* | **pluggable-services** *slot* | **shell-manager** *slot*} [**brief**]

Syntax Description

database	(Optional) Displays database memory information for the specified process.
messaging	(Optional) Displays messaging memory information for specified process. The information displayed is for internal debugging purposes only.
chassis-manager *slot*	Displays memory information for the Chassis Manager process in the specified *slot*. Possible *slot* values are: • **0**—Cisco ASR 1000 Series SPA Interface Processor (SIP) slot 0 • **1**—Cisco ASR 1000 Series SIP slot 1 • **2**—Cisco ASR 1000 Series SIP slot 2 • **f0**—Cisco ASR 1000 Series Embedded Services Processor (ESP) slot 0 • **f1**—Cisco ASR 1000 Series ESP slot 1 • **fp active**—Active Cisco ASR 1000 Series ESP • **fp standby**—Standby Cisco ASR 1000 Series ESP • **r0**—Cisco ASR 1000 Series Route Processor (RP) slot 0 • **r1**—Cisco ASR 1000 Series RP slot 1 • **rp active**—Active Cisco ASR 1000 Series RP • **rp standby**—Standby Cisco ASR 1000 Series RP
cpp-control-process	Displays memory information for the specified Cisco Packet Processor (CPP) Client Control process. Possible *process* values are: • **cpp active**—Active CPP Client Control process • **cpp standby**—Standby CPP Client Control process The information displayed is for internal debugging purposes only.
cpp-driver	Displays memory information for the specified CPP Driver process. Possible *process* values are: • **cpp active**—Active CPPDriver process • **cpp standby**—Standby CPP Driver process The information displayed is for internal debugging purposes only.

cpp-ha-server	Displays memory information for the specified CPP High Availability (HA) Server process. Possible *process* values are: • **cpp active**—Active CPP HA Server process • **cpp standby**—Standby CPP HA Server process The information displayed is for internal debugging purposes only.
cpp-service-process	Displays memory information for the specified CPP Client Service process. Possible *process* values are: • **cpp active**—Active CPP Client Service process • **cpp standby**—Standby CPP Client Service process The information displayed is for internal debugging purposes only.
forwarding-manager *slot*	Displays memory information for the Forwarding Manager process in the specified *slot*. Possible *slot* values are: • **f0**—Cisco ASR 1000 Series ESP slot 0 • **f1**—Cisco ASR 1000 Series ESP slot 1 • **fp active**—Active Cisco ASR 1000 Series ESP • **fp standby**—Standby Cisco ASR 1000 Series ESP • **r0**—Cisco ASR 1000 Series RP slot 0 • **r1**—Cisco ASR 1000 Series RP slot 1 • **rp active**—Active Cisco ASR 1000 Series RP • **rp standby**—Standby Cisco ASR 1000 Series RP
host-manager *slot*	Displays memory information for the Host Manager process in the specified *slot*. Possible *slot* values are: • **0**—Cisco ASR 1000 Series SIP slot 0 • **1**—Cisco ASR 1000 Series SIP slot 1 • **2**—Cisco ASR 1000 Series SIP slot 2 • **f0**—Cisco ASR 1000 Series ESP slot 0 • **f1**—Cisco ASR 1000 Series ESP slot 1 • **fp active**—Active Cisco ASR 1000 Series ESP • **fp standby**—Standby Cisco ASR 1000 Series ESP • **r0**—Cisco ASR 1000 Series RP slot 0 • **r1**—Cisco ASR 1000 Series RP slot 1 • **rp active**—Active Cisco ASR 1000 Series RP • **rp standby**—Standby Cisco ASR 1000 Series RP

interface-manager *slot*	Displays memory information for the Interface Manager process in the specified *slot*. Possible *slot* values are: • **0**—Cisco ASR 1000 Series SIP slot 0 • **1**—Cisco ASR 1000 Series SIP slot 1 • **2**— Cisco ASR 1000 Series SIP slot 2 • **r0**—Cisco ASR 1000 Series RP slot 0 • **r1**—Cisco ASR 1000 Series RP slot 1 • **rp active**—Active Cisco ASR 1000 Series RP • **rp standby**—Standby Cisco ASR 1000 Series RP
ios *slot*	Displays memory information for the IOS process in the specified *slot*. Possible *slot* values are: • **0/0**—Cisco ASR 1000 Series SIP slot 0, bay 0 • **0/1**—Cisco ASR 1000 Series SIP slot 0, bay 1 • **0/2**—Cisco ASR 1000 Series SIP slot 0, bay 2 • **0/3**—Cisco ASR 1000 Series SIP slot 0, bay 3 • **1/0**—Cisco ASR 1000 Series SIP slot 1, bay 0 • **1/1**—Cisco ASR 1000 Series SIP slot 1, bay 1 • **1/2**—Cisco ASR 1000 Series SIP slot 1, bay 2 • **1/3**—Cisco ASR 1000 Series SIP slot 1, bay 3 • **2/0**—Cisco ASR 1000 Series SIP slot 2, bay 0 • **2/1**—Cisco ASR 1000 Series SIP slot 2, bay 1 • **2/2**—Cisco ASR 1000 Series SIP slot 2, bay 2 • **2/3**—Cisco ASR 1000 Series SIP slot 2, bay 3 • **r0**—Cisco ASR 1000 Series RP slot 0 • **r1**—Cisco ASR 1000 Series RP slot 1 • **rp active**—Active Cisco ASR 1000 Series RP • **rp standby**—Standby Cisco ASR 1000 Series RP

logger *slot*	Displays memory information for the logger process in the specified *slot*. Possible *slot* values are: • **0**—Cisco ASR 1000 Series SIP slot 0 • **1**—Cisco ASR 1000 Series SIP slot 1 • **2**—Cisco ASR 1000 Series SIP slot 2 • **f0**—Cisco ASR 1000 Series ESP slot 0 • **f1**—Cisco ASR 1000 Series ESP slot 1 • **fp active**—Active Cisco ASR 1000 Series ESP • **fp standby**—Standby Cisco ASR 1000 Series ESP • **r0**—Cisco ASR 1000 Series RP slot 0 • **r1**—Cisco ASR 1000 Series RP slot 1 • **rp active**—Active Cisco ASR 1000 Series RP • **rp standby**—Standby Cisco ASR 1000 Series RP
pluggable-services *slot*	Displays memory information for the pluggable-services process in the specified *slot*. Possible *slot* values are: • **r0**—Cisco ASR 1000 Series RP slot 0 • **r1**—Cisco ASR 1000 Series RP slot 1 • **rp active**—Active Cisco ASR 1000 Series RP • **rp standby**—Standby Cisco ASR 1000 Series RP
shell-manager *slot*	Displays memory information for the Shell Manager process in the specified slot. Possible *slot* values are: • **r0**—Cisco ASR 1000 Series RP slot 0 • **r1**—Cisco ASR 1000 Series RP slot 1 • **rp active**—Active Cisco ASR 1000 Series RP • **rp standby**—Standby Cisco ASR 1000 Series RP
brief	(Optional) Displays abbreviated memory information for the specified process.

Command Default

No default behavior or values.

Command Modes

Privileged EXEC (#)

Diagnostic (diag)

Command History

Release	Modification
Cisco IOS XE Release 2.1	This command was introduced on the Cisco ASR 1000 Series Routers.

Usage Guidelines

The specification of the **database** and **brief** keywords are optional.

The specification of a process and slot are required.

Examples

The following example displays memory information for the Forwarding Manager process for Cisco ASR 1000 Series RP slot 0:

```
Router# show platform software memory forwarding-manager r0
Module: cdllib
  allocated: 900, requested: 892, overhead: 8
  Allocations: 2, failed: 0, frees: 1
Module: eventutil
  allocated: 117379, requested: 117059, overhead: 320
  Allocations: 46, failed: 0, frees: 6
Module: uipeer
  allocated: 9264, requested: 9248, overhead: 16
  Allocations: 3, failed: 0, frees: 1
Module: Summary
  allocated: 127543, requested: 127199, overhead: 344
  Allocations: 51, failed: 0, frees: 8
```

Table 122 describes the significant fields shown in the display.

Table 122 show platform software memory Field Descriptions

Field	Description
Module:	Name of submodule.
allocated:	Memory, allocated in bytes.
requested:	Number of bytes requested by application.
overhead:	Allocation overhead.
Allocations:	Number of discrete allocation event attempts.
failed:	Number of allocation attempts that were attempted, but failed.
frees:	Number of free events.

The following example displays abbreviated (**brief** keyword) memory information for the Chassis Manager process for Cisco ASR 1000 Series ESP slot 0:

```
Router# show platform software memory chassis-manager f0 brief
  module            allocated     requested     allocs        frees
  ------------------------------------------------------------------------
  CPP Features      692           668           3             0
  Summary           497816        495344        323           14
  chunk             419322        419290        4             0
  eventutil         68546         66146         312           12
  uipeer            9256          9240          4             2
```

Table 123 describes the significant fields shown in the **brief** keyword display.

Table 123 show platform software memory brief Field Descriptions

Field	Description
module	Name of submodule.
allocated	Memory, allocated in bytes.
requested	Number of bytes requested by application.
allocs	Number of discrete allocation event attempts.
frees	Number of free events.

show platform software mount

To display the mounted file systems, both physical and virtual, for a Cisco ASR 1000 Series SPA Interface Processor (SIP), Cisco ASR 1000 Series Embedded Services Processor (ESP), or Cisco ASR 1000 Series Route Processor (RP), use the **show platform software mount** command in privileged EXEC or diagnostic mode.

show platform software mount [*slot* [**brief**]]

Syntax Description

slot	(Optional) Displays mounted file systems for the specified *slot*. Possible *slot* values are: • **0**—Cisco ASR 1000 Series SIP slot 0 • **1**—Cisco ASR 1000 Series SIP slot 1 • **2**—Cisco ASR 1000 Series SIP slot 2 • **f0**—Cisco ASR 1000 Series ESP slot 0 • **f1**—Cisco ASR 1000 Series ESP slot 1 • **fp active**—Active Cisco ASR 1000 Series ESP • **fp standby**—Standby Cisco ASR 1000 Series ESP • **r0**—Cisco ASR 1000 Series RP slot 0 • **r1**—Cisco ASR 1000 Series RP slot 1 • **rp active**—Active Cisco ASR 1000 Series RP • **rp standby**—Standby Cisco ASR 1000 Series RP
brief	(Optional) Displays abbreviated mounted file system information.

Command Default

No default behavior or values.

Command Modes

Privileged EXEC (#)

Diagnostic (diag)

Command History

Release	Modification
Cisco IOS XE Release 2.1	This command was introduced on the Cisco ASR 1000 Series Routers.

Usage Guidelines

If no slot is specified, the command returns mounted file systems for the active RP.

This command allows you to ascertain the presence or absence of specific system mounts. For example, this command might be used to determine /tmp-related mounts, which are used to create many run-time directories and files.

Users may be requested to execute this command to collect information about the underlying configuration of the platform software.

The RP output can differ depending on how the router was booted, and whether there are USB devices inserted.

The SIP and ESP output can differ depending on whether the chassis is a dual or single RP.

Examples

The following example displays mounted file systems for the active RP:

```
Router# show platform software mount
Filesystem                          Used   Available  Use% Mounted on
rootfs                                 0           0     -  /
proc                                   0           0     -  /proc
sysfs                                  0           0     -  /sys
none                                 524     1037640    1%  /dev
/dev/bootflash1                   298263       42410   88%  /bootflash
/dev/harddisk1                    609208     4025132   14%  /misc/scratch
/dev/loop1                         28010           0  100%  /tmp/sw/mount/2007-10-14_...
/dev/loop2                         26920           0  100%  /tmp/sw/mount/2007-10-14_...
/dev/loop3                         48236           0  100%  /tmp/sw/mount/2007-10-14_...
/dev/loop4                          6134           0  100%  /tmp/sw/mount/2007-10-14_...
/dev/loop5                         43386           0  100%  /tmp/sw/mount/2007-10-14_...
/dev/loop6                         30498           0  100%  /tmp/sw/mount/2007-10-14_...
/dev/loop7                         14082           0  100%  /tmp/sw/mount/2007-10-14_...
none                                 524     1037640    1%  /dev
/proc/bus/usb                          0           0     -  /proc/bus/usb
/dev/mtdblock1                       460        1588   23%  /obfl
automount(pid4165)                     0           0     -  /vol
```

The following example displays mounted file systems for the Cisco ASR 1000 Series ESP in ESP slot 0:

```
Router# show platform software mount f0
Filesystem                          Used   Available Use% Mounted on
rootfs                                 0           0     -  /
proc                                   0           0     -  /proc
sysfs                                  0           0     -  /sys
none                               10864      507124    3%  /dev
/dev/loop1                         41418           0  100%  /tmp/sw/fp/0/0/fp/mount
none                               10864      507124    3%  /dev
/proc/bus/usb                          0           0     -  /proc/bus/usb
/dev/mtdblock1                       504        1544   25%  /obfl
automount(pid3210)                     0           0     -  /misc1
```

The following example displays mounted file systems for the active Cisco ASR 1000 Series RP:

```
Router# show platform software mount rp active
Filesystem                          Used   Available  Use% Mounted on
rootfs                                 0           0     -  /
proc                                   0           0     -  /proc
sysfs                                  0           0     -  /sys
none                                 436     1037728    1%  /dev
/dev/bootflash1                   256809       83864   76%  /bootflash
/dev/harddisk1                    252112     4382228    6%  /misc/scratch
/dev/loop1                         30348           0  100%  /tmp/sw/mount/2007-09-27_...
/dev/loop2                         28394           0  100%  /tmp/sw/mount/2007-09-27_...
/dev/loop3                         42062           0  100%  /tmp/sw/mount/2007-09-27_...
/dev/loop4                          8384           0  100%  /tmp/sw/mount/2007-09-27_...
/dev/loop5                         41418           0  100%  /tmp/sw/mount/2007-09-27_...
/dev/loop6                         21612           0  100%  /tmp/sw/mount/2007-09-27_...
/dev/loop7                         16200           0  100%  /tmp/sw/mount/2007-09-27_...
none                                 436     1037728    1%  /dev
/proc/bus/usb                          0           0     -  /proc/bus/usb
```

```
/dev/mtdblock1                          484         1564   24%  /obfl
automount(pid4004)                        0            0     -  /vol
```

Table 124 describes the significant fields shown in the SIP slot (**0**, **1**, or **2**) displays.

Table 124 show platform software mount SIP slot Field Descriptions

Field	Description
Filesystem	Logical name of the file system device.
Used	Number of 1Kb blocks used.
Available	Number of free 1Kb blocks available.
Use%	Percentage of 1Kb blocks used of the total available.
Mounted on	Canonical path to the mounted file system.

The following example displays abbreviated (**brief** keyword) mounted file system information for Cisco ASR 1000 Series SIP slot 0:

```
Router# show platform software mount 0 brief
Mount point: rootfs
  Type     : rootfs
  Location : /
  Options  : rw

Mount point: proc
  Type     : proc
  Location : /proc
  Options  : rw

Mount point: sysfs
  Type     : sysfs
  Location : /sys
  Options  : rw

Mount point: none
  Type     : tmpfs
  Location : /dev
  Options  : rw

Mount point: /dev/loop1
  Type     : iso9660
  Location : /tmp/sw/cc/0/0/cc/mount
  Options  : ro

Mount point: none
  Type     : tmpfs
  Location : /dev
  Options  : rw

Mount point: /proc/bus/usb
  Type     : usbfs
  Location : /proc/bus/usb
  Options  : rw

Mount point: /dev/mtdblock1
  Type     : jffs2
  Location : /obfl
  Options  : rw,noatime,nodiratime
```

```
Mount point: automount(pid3199)
  Type     : autofs
  Location : /misc1
  Options  : rw,fd=5,pgrp=3199,timeout=60,minproto=2,maxproto=4,indirect
```

Table 125 describes the significant fields shown in the **brief** keyword display.

Table 125 show platform software mount brief Field Descriptions

Field	Description
Mount point:	Logical name of the file system device.
Type:	File system type.
Location:	Canonical path to the mounted file system.
Options:	Mount point type-specific flags and settings.

show platform software process list

To display a list of the processes running in a given slot, use the **show platform software process list** command in privileged EXEC or diagnostic mode.

show platform software process list *slot* [**name** *process-name* | **process-id** *process-id* | **summary**]

Syntax Description

slot	Displays running process information for the specified *slot*. Possible *slot* values are: • **0**—Cisco ASR 1000 Series SPA Interface Processor (SIP) slot 0 • **1**—Cisco ASR 1000 Series SIP slot 1 • **2**—Cisco ASR 1000 Series SIP slot 2 • **f0**—Cisco ASR 1000 Series Embedded Services Processor (ESP) slot 0 • **f1**—Cisco ASR 1000 Series ESP slot 1 • **fp active**—Active Cisco ASR 1000 Series ESP • **fp standby**—Standby Cisco ASR 1000 Series ESP • **r0**—Cisco ASR 1000 Series Route Processor (RP) slot 0 • **r1**—Cisco ASR 1000 Series RP slot 1 • **rp active**—Active Cisco ASR 1000 Series RP • **rp standby**—Standby Cisco ASR 1000 Series RP
name *process-name*	(Optional) Displays information for the specified process name.
process-id *process-id*	(Optional) Displays information for the specified process ID.
summary	(Optional) Displays summary process information for the running host.

Command Default

No default behavior or values.

Command Modes

Privileged EXEC (#)

Diagnostic (diag)

Command History

Release	Modification
Cisco IOS XE Release 2.1	This command was introduced on the Cisco ASR 1000 Series Routers.

Usage Guidelines

The **name** and **process-id** keywords can be used to narrow the process list display down to specific processes.

The **summary** keyword can be used to display summary information about running processes.

Examples

The following example displays information about running processes for Cisco ASR 1000 Series SIP slot 0:

```
Router# show platform software process list 0
Name                       Pid    PPid  Group Id  Status    Priority  Size
------------------------------------------------------------------------------
init                         1       0         1  S               20  1974272
ksoftirqd/0                  2       1         1  S               39  0
events/0                     3       1         1  S               15  0
khelper                      4       1         1  S               15  0
kthread                      5       1         1  S               15  0
kblockd/0                   19       5         1  S               15  0
khubd                       23       5         1  S               15  0
pdflush                     59       5         1  S               20  0
pdflush                     60       5         1  S               20  0
kswapd0                     61       5         1  S               15  0
aio/0                       62       5         1  S               15  0
xfslogd/0                   63       5         1  S               15  0
xfsdatad/0                  64       5         1  S               15  0
mtdblockd                  626       1         1  S               20  0
loop0                     1370       1         1  S                0  0
portmap                   1404       1      1404  S               20  2076672
portmap                   1406       1      1406  S               20  2076672
loop1                     1440       1         1  S                0  0
udevd                     2104       1      2104  S               16  1974272
jffs2_gcd_mtd1            2796       1         1  S               30  0
klogd                     3093       1      3093  S               20  1728512
automount                 3199       1      3199  S               20  2396160
xinetd                    3214       1      3214  S               20  3026944
xinetd                    3216       1      3216  S               20  3026944
pvp.sh                    3540       1      3540  S               20  3678208
inotifywait               3575    3540      3575  S               20  1900544
pman.sh                   3614    3540      3614  S               20  3571712
pman.sh                   3714    3540      3714  S               20  3571712
btrace_rotate.s           3721    3614      3721  S               20  3133440
agetty                    3822       1      3822  S               20  1720320
mcp_chvrf.sh              3823       1      3823  S               20  2990080
sntp                      3824       1      3824  S               20  2625536
issu_switchover           3825       1      3825  S               20  3899392
xinetd                    3827    3823      3823  S               20  3026944
cmcc                      3862    3714      3862  S               20  26710016
pman.sh                   3883    3540      3883  S               20  3571712
pman.sh                   4014    3540      4014  S               20  3575808
hman                      4020    3883      4020  R               20  19615744
imccd                     4114    4014      4114  S               20  31539200
inotifywait               4196    3825      3825  S               20  1896448
pman.sh                   4351    3540      4351  S               20  3575808
plogd                     4492    4351      4492  S               20  22663168
inotifywait               4604    3721      4604  S               20  1900544
```

Table 126 describes the significant fields shown in the display.

Table 126 show platform software process list Field Descriptions

Field	Description
Name	Name of the process.
Pid	Process ID.
PPid	Parent Process ID.
Group Id	Process group ID.

Table 126 show platform software process list Field Descriptions (continued)

Field	Description
Status	Process status.
Priority	Process priority.
Size	Virtual memory size (in bytes).

The following example displays information about a specific named process for Cisco ASR 1000 Series SIP slot 0:

```
Router# show platform software process list 0 name sleep
Name: sleep
  Process id       : 25938
  Parent process id: 3891
  Group id         : 3891
  Status           : S
  Session id       : 3816
  User time        : 0
  Kernel time      : 0
  Priority         : 20
  Virtual bytes    : 2482176
  Resident pages   : 119
  Resident limit   : 4294967295
  Minor page faults: 182
  Major page faults: 0
```

The following example displays information about a specific process identifier for Cisco ASR 1000 Series SIP slot 0:

```
Router# show platform software process list 0 process-id 1
Name: init
  Process id       : 1
  Parent process id: 0
  Group id         : 1
  Status           : S
  Session id       : 1
  User time        : 1
  Kernel time      : 741
  Priority         : 20
  Virtual bytes    : 1974272
  Resident pages   : 161
  Resident limit   : 4294967295
  Minor page faults: 756
  Major page faults: 0
```

Table 127 describes the significant fields shown in the **name** and **process-id** keyword displays.

Table 127 show platform software process list name and process-id Field Descriptions

Field	Description
Name	Name of the process.
Process id	Process ID.
Parent process id	Parent process ID.
Group id	Process group ID.
Status	Process status.

Table 127 show platform software process list name and process-id Field Descriptions (continued)

Field	Description
Session id	Process session ID.
User time	Time (in seconds) spent in user mode.
Kernel time	Time (in seconds) spent in kernel mode.
Priority	Process priority.
Virtual bytes	Virtual memory size (in bytes).
Resident pages	Resident page size.
Resident limit	Current limit on Resident pages.
Minor page faults	Number of minor page faults.
Major page faults	Number of major page faults.

The following example displays process summary information for Cisco ASR 1000 Series SIP slot 0:

```
Router# show platform software process list 0 summary
Total number of processes: 54
  Running          : 4
  Sleeping         : 50
  Disk sleeping    : 0
  Zombies          : 0
  Stopped          : 0
  Paging           : 0

  Up time          : 1562
  Idle time        : 1511
  User time        : 1606
  Kernel time      : 1319

  Virtual memory   : 587894784
  Pages resident   : 45436
  Major page faults: 25
  Minor page faults: 149098

Architecture     : ppc
  Memory (kB)
    Physical       : 524288
    Total          : 479868
    Used           : 434948
    Free           : 44920
    Active         : 183020
    Inactive       : 163268
    Inact-dirty    : 0
    Inact-clean    : 0
    Dirty          : 0
    AnonPages      : 76380
    Bounce         : 0
    Cached         : 263764
    Commit Limit   : 239932
    Committed As   : 201452
    High Total     : 0
    High Free      : 0
    Low Total      : 479868
    Low Free       : 44920
    Mapped         : 59996
    NFS Unstable   : 0
    Page Tables    : 1524
```

```
  Slab           : 73760
  VMmalloc Chunk : 426840
  VMmalloc Total : 474856
  VMmalloc Used  : 47372
  Writeback      : 0

Swap (kB)
  Total          : 0
  Used           : 0
  Free           : 0
  Cached         : 0

Buffers (kB)     : 6144

Load Average
  1-Min          : 0.00
  5-Min          : 0.00
  15-Min         : 0.00
```

Table 128 describes the significant fields shown in the **summary** keyword display.

Table 128 show platform software process list summary Field Descriptions

Field	Description
Total number of processes	Total number of processes in all possible states.
Running	Number of processes in the running state.
Sleeping	Number of processes in the sleeping state.
Disk sleeping	Number of processes in the disk-sleeping state.
Zombies	Number of processes in the zombie state.
Stopped	Number of processes in the stopped state.
Paging	Number of processes in the paging state.
Up time	System Up time (in seconds).
Idle time	System Idle time (in seconds).
User time	System time (in seconds) spent in user mode.
Kernel time	System time (in seconds) spent in kernel mode.
Virtual memory	Virtual memory size (in bytes).
Pages resident	Resident page size.
Major page faults	Number of major page faults.
Minor page faults	Number of minor page faults.
Architecture	System CPU architecture: PowerPC (ppc).
Memory (kB)	System memory heading.
Physical	Total physical memory (in kilobytes).
Total	Total available memory (in kilobytes). This value represents the physical memory available for kernel use.
Used	Used memory (in kilobytes).
Free	Free memory (in kilobytes).
Active	Most recently used memory (in kilobytes).

Table 128 show platform software process list summary Field Descriptions (continued)

Field	Description
Inactive	Memory (in kilobytes) that has been less recently used. It is more eligible to be reclaimed for other purposes.
Inact-dirty	Memory (in kilobytes) that may need to be written to persistent store (cache or disk).
Inact-clean	Memory (in kilobytes) that is readily available for re-use.
Dirty	Memory (in kilobytes) that is waiting to get written back to the disk.
AnonPages	Memory (in kilobytes) that is allocated when a process requests memory from the kernel via the malloc() system call. This memory has no file backing on disk.
Bounce	Memory (in kilobytes) that is allocated to bounce buffers.
Cached	Amount of physical RAM (in kilobytes) used as cache memory.
Commit Limit	Total amount of memory (in kilobytes) currently available to be allocated on the system. This limit is only adhered to if strict overcommit accounting is enabled.
Committed As	Total amount of memory (in kilobytes) presently allocated on the system. The committed memory is a sum of all of the memory that has been allocated by processes, even if it has not been used by them as of yet.
High Total	Total amount of memory (in kilobytes) that is not directly mapped into kernel space. The High Total value can vary based on the type of kernel used.
High Free	Amount of free memory (in kilobytes) that is not directly mapped into kernel space. The High Free value can vary based on the type of kernel used.
Low Total	Total amount of memory (in kilobytes) that is directly mapped into kernel space. The Low Total value can vary based on the type of kernel used.
Low Free	Amount of free memory (in kilobytes) that is directly mapped into kernel space. The Low Free value can vary based on the type of kernel used.
Mapped	Total amount of memory (in kilobytes) that has been used to map devices, files, or libraries using the **mmap** command.
NFS Unstable	Total amount of memory (in kilobytes) used for unstable NFS pages. Unstable NFS pages are pages that have been written into the page cache on the server, but have not yet been synchronized to disk.
Page Tables	Total amount of memory (in kilobytes) dedicated to the lowest page table level.
Slab	Total amount of memory (in kilobytes) used by the kernel to cache data structures for its own use.

Table 128 show platform software process list summary Field Descriptions (continued)

Field	Description
VMalloc Chunk	Largest contiguous block of available virtual address space (in kilobytes) that is free.
VMalloc Total	Total amount of memory (in kilobytes) of total allocated virtual address space.
VMalloc Used	Total amount of memory (in kilobytes) of used virtual address space.
Writeback	Memory (in kilobytes) that is actively being written back to the disk.
Swap (kB)	Swap memory heading.
Total	Total swap memory (in kilobytes).
Used	Used swap memory (in kilobytes).
Free	Free swap memory (in kilobytes).
Cached	Cached swap memory (in kilobytes).
Buffers (kB)	Buffers heading.
Load Average	Indicators of system load.
1-Min	Average number of processes running for the last minute.
5-Min	Average number of processes running for the last 5 minutes.
15-Min	Average number of processes running for the last 15 minutes.

show platform software tech-support

To display system information or create a technical support information tar file for Cisco Technical Support, use the **show platform software tech-support** command in privileged EXEC or diagnostic mode.

show platform software tech-support [**file** {**bootflash:***filename*.**tgz** | **fpd:***filename*.**tgz** | **harddisk:***filename*.**tgz** | **obfl:***filename*.**tgz** | **stby-bootflash:***filename*.**tgz** | **stby-harddisk:***filename*.**tgz** | **stby-obfl:***filename*.**tgz** | **stby-usb0:***filename*.**tgz** | **stby-usb1:***filename*.**tgz**}]

Syntax Description

file	(Optional) Creates a technical support information tar file for the specified destination file path.
bootflash:*filename*.**tgz**	Creates a technical support information tar file for the boot flash memory file system on the active RP.
fpd:*filename*.**tgz**	Creates a technical support information tar file for the field-programmable device (FPD) image package on the active RP. The information displayed is for internal debugging puposes only.
harddisk:*filename*.**tgz**	Creates a technical support information tar file for the hard disk file system on the active RP.
obfl:*filename*.**tgz**	Creates a technical support information tar file for the file system for Onboard Failure Logging (obfl) files. The information displayed is for internal debugging puposes only.
stby-bootflash: *filename*.**tgz**	Creates a technical support information tar file for the boot flash memory file system on the standby RP. The information displayed is for internal debugging puposes only.
stby-harddisk: *filename*.**tgz**	Creates a technical support information tar file for the hard disk file system on the standby RP. The information displayed is for internal debugging puposes only.
stby-obfl:*filename*.**tgz**	Creates a technical support information tar file for the Onboard Failure Logging (obfl) files on the standby RP. The information displayed is for internal debugging puposes only.
stby-usb0:*filename*.**tgz**	Creates a technical support information tar file for Universal Serial Bus (USB) memory. The information displayed is for internal debugging puposes only.
stby-usb1:*filename*.**tgz**	Creates a technical support information tar file for Universal Serial Bus (USB) memory. The information displayed is for internal debugging puposes only.

Command Default

No default behavior or values.

Command Modes

Privileged EXEC (#)

Diagnostic (diag)

Command History

Release	Modification
Cisco IOS XE Release 2.1	This command was introduced on the Cisco ASR 1000 Series Routers.

Usage Guidelines

If the **file** keyword is specified, the specification of the **bootflash:** or **harddisk:** keyword and filename is required.

The **show platform software tech-support** command without a destination file path specification returns a large volume of information in a short period of time. You should save the output of the **show platform software tech-support** command in a log file to send to Cisco Technical Support for analysis.

Examples

The following example displays system information for Cisco Technical Support:

```
Router# show platform software tech-support
---- show version installed -----
Type: provisioning file, Version: unknown
  Provisioned on: RP0, Status: active
  File: packages.conf.super
  Modified: 2007-11-07 15:06:12.212303000 +0000
  SHA1 (header): d929d995d5ba2d3dedf67137c3e0e321b1727d7b
  SHA1 (calculated): d929d995d5ba2d3dedf67137c3e0e321b1727d7b
  SHA1 (external): a16881b6a7e3a5593b63bf211f72b8af9c534063
instance address        : 0X890DE9B4
    fast failover address   : 00000000
    cpp interface handle  0
    instance address        : 0X890DE9B8
    fast failover address   : 00000000
    cpp interface handle  0
    instance address        : 0X890DE9BC
    fast failover address   : 00000000
...
```

Note The **show platform software tech-support** command returns a large volume of information in a short period of time. The example above has been abbreviated for the purposes of this description.

The following example creates a technical support information tar file for the boot flash memory file system on the active RP:

```
Router# show platform software tech-support file bootflash:tech_support_output.tgz
Running tech support command set; please wait...
Creating file 'bootflash:target_support_output.tgz.tgz' ...
File 'bootflash:target_support_output.tgz.tgz' created successfully
```

The following example creates a technical support information tar file for the hard disk file system on the active RP:

```
Router# show platform software tech-support file harddisk:tech_support_output.tgz
Running tech support command set; please wait...
Creating file 'harddisk:tech_support_ouput.tgz.tgz' ...
File 'harddisk:tech_support_ouput.tgz.tgz' created successfully
```

show power

To display information about the power status, use the **show power** command in user EXEC or privileged EXEC mode**.**

show power [**available** | **inline** [*interface number* | **module number**] | **redundancy-mode** | **status** {**all** | **fan-tray** *fan-tray-number* | **module** *slot* | **power-supply** *pwr-supply-number*} | **total** | **used**]

Syntax Description

available	(Optional) Displays the available system power (margin).
inline	(Optional) Displays the inline power status.
interface number	(Optional) Specifies the interface type; possible valid values are **ethernet**, **fastethernet**, **gigabitethernet**, **tengigabitethernet**, **null**, **port-channel**, and **vlan**. See the "Usage Guidelines" section for additional information.
module number	Displays the power status for a specific module.
redundancy-mode	(Optional) Displays the power-supply redundancy mode.
status	(Optional) Displays the power status.
all	Displays all the FRU types.
fan-tray *fan-tray-number*	Displays the power status for the fan tray.
module *slo*t	Displays the power status for a specific module.
power-supply *pwr-supply-number*	Displays the power status for a specific power supply; valid values are **1** and **2**.
total	(Optional) Displays the total power that is available from the power supplies.
used	(Optional) Displays the total power that is budgeted for powered-on items.

Defaults

This command has no default settings.

Command Modes

User EXEC
Privileged EXEC

Command History

Release	Modification
12.2(14)SX	Support for this command was introduced on the Supervisor Engine 720.
12.2(17a)SX1	The output was changed to include the total system-power information.
12.2(17b)SXA	This command was changed to include information about the inline power status for a specific module.
12.2(17d)SXB	Support for this command on the Supervisor Engine 2 was extended to Release 12.2(17d)SXB.

Release	Modification
12.2(18)SXF	The output was changed to include information about the high-capacity power supplies.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

The *interface-number* argument designates the module and port number. Valid values for *interface-number* depend on the specified interface type and the chassis and module that are used. For example, if you specify a Gigabit Ethernet interface and have a 48-port 10/100BASE-T Ethernet module that is installed in a 13-slot chassis, valid values for the module number are from 1 to 13 and valid values for the port number are from 1 to 48.

Valid values for *vlan-id* are from 1 to 4094.

The Inline power field in the **show power** output displays the inline power that is consumed by the modules. For example, this example shows that module 9 has consumed 0.300 A of inline power:

```
Inline power   #    current
module         9    0.300A
```

Examples

This example shows how to display the available system power:

```
Router> show power available

system power available = 20.470A
Router>
```

This example shows how to display power-supply redundancy mode:

```
Router# show power redundancy-mode

system power redundancy mode = redundant
Router#
```

This command shows how to display the system-power status:

```
Router> show power

system power redundancy mode = combined
system power total =     3984.12 Watts (94.86 Amps @ 42V)
system power used =      1104.18 Watts (26.29 Amps @ 42V)
system power available = 2879.94 Watts (68.57 Amps @ 42V)
                       Power-Capacity PS-Fan Output Oper
PS   Type              Watts   A @42V Status Status State
---- ----------------- ------- ------ ------ ------ -----
1    WS-CAC-3000W      2830.80 67.40  OK     OK     on
2    WS-CAC-1300W      1153.32 27.46  OK     OK     on
Note: PS2 capacity is limited to 2940.00 Watts (70.00 Amps @ 42V)
      when PS1 is not present
                       Pwr-Allocated  Oper
Fan  Type              Watts   A @42V State
---- ----------------- ------- ------ -----
1    FAN-MOD-9          241.50  5.75  OK
2                       241.50  5.75  failed
                       Pwr-Requested  Pwr-Allocated  Admin Oper
Slot Card-Type         Watts   A @42V Watts   A @42V State State
---- ----------------- ------- ------ ------- ------ ----- -----
1    WS-X6K-SUP2-2GE    145.32  3.46   145.32  3.46  on    on
2                         -     -      145.32  3.46  -     -
```

```
3    WS-X6516-GBIC       118.02  2.81   118.02  2.81  on    on
5    WS-C6500-SFM        117.18  2.79   117.18  2.79  on    on
7    WS-X6516A-GBIC      214.20  5.10     -     -     on    off (insuff cooling capacity)
8    WS-X6516-GE-TX      178.50  4.25   178.50  4.25  on    on
9    WS-X6816-GBIC       733.98 17.48     -     -     on    off (connector rating
exceeded)
Router>
```

This example shows how to display the power status for all FRU types:

```
Router# show power status all

FRU-type       #    current   admin state oper
power-supply   1    27.460A   on          on
module         1    4.300A    on          on
module         2    4.300A    -           -   (reserved)
module         5    2.690A    on          on
Router#
```

This example shows how to display the power status for a specific module:

```
Router# show power status module 1

FRU-type       #    current   admin state oper
module         1    -4.300A   on          on
Router#
```

This example shows how to display the power status for a specific power supply:

```
Router# show power status power-supply 1

FRU-type       #    current   admin state oper
power-supply   1    27.460A   on          on
Router#
```

This example displays information about the high-capacity power supplies:

```
Router# show power status power-supply 2

                       Power-Capacity PS-Fan Output Oper
PS   Type              Watts   A @42V Status Status State
---- ----------------- ------- ------ ------ ------ -----
1    WS-CAC-6000W      2672.04  63.62 OK     OK     on
2    WS-CAC-9000W-E    2773.68  66.04 OK     OK     on
Router#
```

This example shows how to display the total power that is available from the power supplies:

```
Router# show power total

system power total = 27.460A
Router#
```

This example shows how to display the total power that is budgeted for powered-on items:

```
Router# show power used

system power used = -6.990A
Router#
```

This command shows how to display the inline power status on the interfaces:

```
Router# show power inline

Interface            Admin    Oper    Power ( mWatt )  Device
-------------------- ----- ---------- --------------- -----------
```

```
FastEthernet9/1      auto  on         6300            Cisco 6500 IP Phone
FastEthernet9/2      auto  on         6300            Cisco 6500 IP Phone
.
.
. <Output truncated>
```

This command shows how to display the inline power status for a specific module:

```
Router# show power inline mod 7

Interface  Admin    Oper    Power       Device        Class
                            (Watts)
---------- ----- ---------- ------- --------------  -----------
Gi7/1      auto   on            6.3  Cisco IP Phone 7960  n/a
Gi7/2      static power-deny      0  Ieee PD              3
.
.
. <Output truncated>
```

Related Commands

Command	Description
power enable	Turns on power for the modules.
power redundancy-mode	Sets the power-supply redundancy mode.

show processes

To display information about the active Cisco IOS processes or the Cisco IOS Software Modularity POSIX-style processes, use the **show processes** command in privileged EXEC mode.

Cisco IOS Software

show processes [**history** | *process-id*]

Cisco IOS Software Modularity

show processes

Syntax Description

history	(Optional) For Cisco IOS processes only. Displays the process history in an ordered format.
process-id	(Optional) For Cisco IOS processes only. An integer that specifies the process for which memory and CPU utilization data shall be returned.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
10.0	This command was introduced.
12.2(2)T	The **history** keyword was added.
12.3(2)T	The *process-id* argument was added.
12.2(18)SXF4	The syntax was modified to support Cisco IOS Software Modularity images.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

Cisco IOS Software Modularity

Although no optional keywords or arguments are supported for the base **show processes** command when a Software Modularity image is running, more details about processes are displayed using the **show processes cpu**, **show processes detailed**, **show processes kernel**, and **show processes memory** commands.

Examples

Example output varies between Cisco IOS software images and Cisco IOS Software Modularity software images. To view the appropriate output, choose one of the following sections:

- Cisco IOS Software
- Cisco IOS Software Modularity

Cisco IOS Software

The following is sample output from the **show processes** command:

```
Router# show processes
```

```
CPU utilization for five seconds: 21%/0%; one minute: 2%; five minutes: 2%
 PID QTy       PC Runtime (ms)    Invoked   uSecs    Stacks TTY Process
   1 Cwe 606E9FCC           0          1       0 5600/6000   0 Chunk Manager
   2 Csp 607180F0           0     121055       0 2608/3000   0 Load Meter
   3 M*         0           8         90      88 9772/12000  0 Exec
   4 Mwe 619CB674           0          1       023512/24000  0 EDDRI_MAIN
   5 Lst 606F6AA4       82064      61496    1334 5668/6000   0 Check heaps
   6 Cwe 606FD444           0        127       0 5588/6000   0 Pool Manager
   7 Lwe 6060B364           0          1       0 5764/6000   0 AAA_SERVER_DEADT
   8 Mst 6063212C           0          2       0 5564/6000   0 Timers
   9 Mwe 600109D4           0          2       0 5560/6000   0 Serial Backgroun
  10 Mwe 60234848           0          2       0 5564/6000   0 ATM Idle Timer
  11 Mwe 602B75F0           0          2       0 8564/9000   0 ATM AutoVC Perio
  12 Mwe 602B7054           0          2       0 5560/6000   0 ATM VC Auto Crea
  13 Mwe 606068B8           0          2       0 5552/6000   0 AAA high-capacit
  14 Msi 607BABA4      251264     605013     415 5628/6000   0 EnvMon
  15 Mwe 607BFF8C           0          1       0 8600/9000   0 OIR Handler
  16 Mwe 607D407C           0      10089       0 5676/6000   0 IPC Dynamic Cach
  17 Mwe 607CD03C           0          1       0 5632/6000   0 IPC Zone Manager
  18 Mwe 607CCD80           0     605014       0 5708/6000   0 IPC Periodic Tim
  19 Mwe 607CCD24           0     605014       0 5704/6000   0 IPC Deferred Por
  20 Mwe 607CCE2C           0          1       0 5596/6000   0 IPC Seat Manager
```

Table 129 describes the fields shown in the display.

Table 129 show processes Field Descriptions

Field	Description
CPU utilization for five seconds	CPU utilization for the last 5 seconds. The second number indicates the percentage of CPU time spent at the interrupt level.
one minute	CPU utilization for the last minute.
five minutes	CPU utilization for the last 5 minutes.
PID	Process ID.
Q	Process queue priority. Possible values: C (critical), H (high), M (medium), and L (low).
Ty	Scheduler test. Possible values: * (currently running), E (waiting for an event), S (ready to run, voluntarily relinquished processor), rd (ready to run, wakeup conditions have occurred), we (waiting for an event), sa (sleeping until an absolute time), si (sleeping for a time interval), sp (sleeping for a time interval as an alternate call, st (sleeping until a timer expires), hg (hung: the process will never execute again), xx (dead: the process has terminated, but has not yet been deleted).
PC	Current program counter.
Runtime (ms)	CPU time that the process has used (in milliseconds).
Invoked	Number of times that the process has been invoked.
uSecs	Microseconds of CPU time for each process invocation.
Stacks	Low water mark/Total stack space available (in bytes).
TTY	Terminal that controls the process.
Process	Name of the process.

Note Because platforms have a 4- to 8-millisecond clock resolution, run times are considered reliable only after a large number of invocations or a reasonable, measured run time.

For a list of process descriptions, see http://www.cisco.com/warp/public/63/showproc_cpu.html.

The following is sample output from the **show processes history** command:

```
Router# show processes history

PID Exectime(ms) Caller PC Process Name
   3           12 0x0        Exec
  16            0 0x603F4DEC GraphIt
  21            0 0x603CFEF4 TTY Background
  22            0 0x6042FD7C Per-Second Jobs
  67            0 0x6015CD38 SMT input
  39            0 0x60178804 FBM Timer
  16            0 0x603F4DEC GraphIt
  21            0 0x603CFEF4 TTY Background
  22            0 0x6042FD7C Per-Second Jobs
  16            0 0x603F4DEC GraphIt
  21            0 0x603CFEF4 TTY Background
  22            0 0x6042FD7C Per-Second Jobs
  67            0 0x6015CD38 SMT input
  39            0 0x60178804 FBM Timer
  24            0 0x60425070 Compute load avgs
  11            0 0x605210A8 ARP Input
  69            0 0x605FDAF4 DHCPD Database
  69            0 0x605FD568 DHCPD Database
  51            0 0x60670B3C IP Cache Ager
  69            0 0x605FD568 DHCPD Database
  36            0 0x606E96DC SSS Test Client
  69            0 0x605FD568 DHCPD Database
 --More--
```

Table 130 describes the significant fields shown in the display.

Table 130 show processes history Field Descriptions

Field	Description
PID	Process ID.
Exectime (ms)	Execution time of the most recent run or the total execution time of the most recent consecutive runs.
Caller PC	Current program counter of this process before it was suspended.
Process Name	Name of the process.

The following is sample output from the **show processes** *process-id* command:

```
Router# show processes 6

Process ID 6 [Pool Manager], TTY 0
Memory usage [in bytes]
 Holding: 921148, Maximum: 940024, Allocated: 84431264, Freed: 99432136
 Getbufs: 0, Retbufs: 0, Stack: 12345/67890
CPU usage
 PC: 0x60887600, Invoked: 188, Giveups: 100, uSec: 24
 5Sec: 3.03%, 1Min: 2.98%, 5Min: 1.55%, Average: 0.58%,
```

```
Age: 662314 msec, Runtime: 3841 msec
State: Running, Priority: Normal
```

Table 131 describes the fields shown in the display.

Table 131 show processes process-id Field Descriptions

Field	Description
Process ID	Process ID number and process name.
TTY	Terminal that controls the process.
Memory usage [in bytes]	This section contains fields that show the memory used by the specified process.
Holding	Amount of memory currently allocated to the process.
Maximum	Maximum amount of memory allocated to the process since its invocation.
Allocated	Bytes of memory allocated by the process.
Freed	Bytes of memory freed by the process.
Getbufs	Number of times that the process has requested a packet buffer.
Retbufs	Number of times that the process has relinquished a packet buffer.
Stack	Low water mark/Total stack space available (in bytes).
CPU usage	This section contains fields that show the CPU resources used by the specified process.
PC	Current program counter of this process before it was suspended.
Invoked	Number of times that the process executed since its invocation.
Giveups	Number of times that the process voluntarily gave up the CPU.
uSec	Microseconds of CPU time for each process invocation.
5Sec	CPU utilization by process in the last five seconds.
1Min	CPU utilization by process in the last minute.
5Min	CPU utilization by process in the last five minutes.
Average	The average amount of CPU utilization by the process since its invocation.
Age	Milliseconds since the process was invoked.
Runtime	CPU time that the process has used (in milliseconds).
State	Current state of the process. Possible values: Running, Waiting for Event, Sleeping (Mgd Timer), Sleeping (Periodic), Ready, Idle, Dead.
Priority	The priority of the process. Possible values: Low, Normal, High.

Cisco IOS Software Modularity

The following is sample output from the **show processes** command when a Cisco IOS Software Modularity image is running:

```
Router# show processes

Total CPU utilization for 5 seconds: 99.7%; 1 minute: 98.9%; 5 minutes: 86.5%
PID   TID  Prio STATE       Blocked   Stack           CPU  Name
1     1    0    Ready                0    (128K)   2m28s  procnto-cisco
1     2    63   Receive     1        0    (128K)   0.000  procnto-cisco
1     3    10   Receive     1        0    (128K)   0.000  procnto-cisco
```

```
1     4    11   Receive     1          0    (128K)   1.848  procnto-cisco
1     5    63   Receive     1          0    (128K)   0.000  procnto-cisco
1     6    63   Receive     1          0    (128K)   0.000  procnto-cisco
12290 1    10   Receive     1          12288(128K)   0.080  chkptd.proc
12290 2    10   Receive     8          12288(128K)   0.000  chkptd.proc
3     1    15   Condvar     1027388    12288(128K)   0.016  qdelogger
3     2    15   Receive     1          12288(128K)   0.004  qdelogger
3     3    16   Condvar     1040024    12288(128K)   0.004  qdelogger
4     1    10   Receive     1          4096 (128K)   0.016  devc-pty
6     1    62   Receive     1          8192 (128K)   0.256  devc-ser2681
6     2    63   Intr                   8192 (128K)   0.663  devc-ser2681
7     1    10   Receive     1          32768(128K)   0.080  dumper.proc
7     2    10   Receive     1          32768(128K)   0.008  dumper.proc
7     3    10   Receive     1          32768(128K)   0.000  dumper.proc
7     4    10   Receive     1          32768(128K)   0.020  dumper.proc
7     5    10   Receive     1          32768(128K)   0.008  dumper.proc
4104  2    10   Receive     1          12288(128K)   0.000  pipe
4104  3    10   Receive     1          12288(128K)   0.000  pipe
8210  1    10   Nanosleep              8192 (128K)   0.040  watchdog.proc
8211  1    10   Receive     1          16384(128K)   0.044  syslogd.proc
8211  2    10   Receive     7          16384(128K)   0.000  syslogd.proc
8211  3    10   Sigwaitin              16384(128K)   0.000  syslogd.proc
8212  2    10   Receive     1          24576(128K)   0.024  name_svr.proc
8212  3    10   Receive     1          24576(128K)   0.100  name_svr.proc
8212  4    10   Receive     1          24576(128K)   0.340  name_svr.proc
8212  5    10   Receive     1          24576(128K)   0.304  name_svr.proc
8213  1    10   Receive     1          24576(128K)   0.644  wdsysmon.proc
8213  2    10   Receive     5          24576(128K)   0.052  wdsysmon.proc
8213  3    10   Receive     10         24576(128K)   0.004  wdsysmon.proc
8213  4    63   Nanosleep              24576(128K)   0.000  wdsysmon.proc
8214  1    10   Receive     1          94208(128K)   0.132  sysmgr.proc
8214  2    10   Sigwaitin              94208(128K)   0.000  sysmgr.proc
8214  3    10   Receive     8          94208(128K)   0.004  sysmgr.proc
8214  4    10   Receive     1          94208(128K)   0.000  sysmgr.proc
8214  5    10   Receive     1          94208(128K)   0.000  sysmgr.proc
8214  6    10   Receive     1          94208(128K)   0.004  sysmgr.proc
8214  7    10   Receive     1          94208(128K)   0.000  sysmgr.proc
8214  8    10   Receive     1          94208(128K)   0.000  sysmgr.proc
8214  9    10   Receive     1          94208(128K)   0.000  sysmgr.proc
8214  10   10   Receive     1          94208(128K)   0.000  sysmgr.proc
12317 1    10   Receive     23         73728(128K)   2.212  ios-base
12317 2    10   Receive     1          73728(128K)   0.064  ios-base
12317 3    10   Reply       1          73728(128K)  17.800  ios-base
12317 4    11   Nanosleep              73728(128K)   0.000  ios-base
12317 5    10   Receive     1          73728(128K)  21.108  ios-base
12317 6    45   Intr                   73728(128K)   0.000  ios-base
12317 7    35   Intr                   73728(128K)   0.064  ios-base
12317 8    10   Reply       12336      73728(128K)   0.776  ios-base
12317 9    10   Receive     1          73728(128K)  12.608  ios-base
12317 10   25   Intr                   73728(128K)  26.404  ios-base
12317 11   25   Intr                   73728(128K)   0.088  ios-base
12317 12   45   Intr                   73728(128K)   0.000  ios-base
12317 13   10   Receive     1          73728(128K)   6.456  ios-base
12317 14   20   Reply       6          73728(128K)   0.064  ios-base
12317 15   10   Receive     1          73728(128K)   8.064  ios-base
12324 1    10   Receive     1          40960(128K)  73.088  iprouting.iosproc
12324 2    10   Ready                  40960(128K)  32.552  iprouting.iosproc
12324 4    11   Nanosleep              40960(128K)   0.000  iprouting.iosproc
12324 5    10   Receive     1          40960(128K)   4.312  iprouting.iosproc
12324 6    10   Receive     1          40960(128K)   6.988  iprouting.iosproc
12324 7    10   Reply       1          40960(128K)  41.108  iprouting.iosproc
12324 8    10   Receive     1          40960(128K)   0.032  iprouting.iosproc
12324 9    10   Reply       1          40960(128K)   0.332  iprouting.iosproc
12330 1    10   Receive     1          36864(128K)   0.000  cdp2.iosproc
```

```
12330 2    10   Receive     1          36864(128K)   0.004  cdp2.iosproc
12330 3    10   Receive     1          36864(128K)   0.024  cdp2.iosproc
12330 4    11   Nanosleep               36864(128K)   0.000  cdp2.iosproc
12330 5    10   Reply       1          36864(128K)   0.228  cdp2.iosproc
12330 6    10   Receive     1          36864(128K)   0.000  cdp2.iosproc
12330 7    10   Receive     9          36864(128K)   0.000  cdp2.iosproc
12334 1    10   Receive     1          45056(128K)   0.000  inetd.proc
12334 2    10   Sigwaitin               45056(128K)   0.000  inetd.proc
12334 3    10   Receive     1          45056(128K)   0.000  inetd.proc
12334 4    10   Receive     1          45056(128K)   0.020  inetd.proc
12334 5    10   Receive     1          45056(128K)   0.000  inetd.proc
12335 1    10   Receive     1          118784(128K)   0.000  tcp.proc
12335 2    10   Receive     1          118784(128K)   0.000  tcp.proc
12335 3    10   Sigwaitin               118784(128K)   0.000  tcp.proc
12335 4    10   Condvar     7A602080   118784(128K)   5.092  tcp.proc
12335 5    10   Ready                  118784(128K)  21.092  tcp.proc
12335 6    10   Receive     1          118784(128K)  14.280  tcp.proc
12335 7    10   Receive     1          118784(128K)   0.000  tcp.proc
12336 1    10   Receive     1          53248(128K)   0.000  udp.proc
12336 3    10   Sigwaitin               53248(128K)   0.000  udp.proc
12336 4    10   Condvar     7A602080   53248(128K)   0.000  udp.proc
12336 5    10   Receive     11         53248(128K)   0.072  udp.proc
12336 6    10   Receive     1          53248(128K)   0.028  udp.proc
12336 7    10   Receive     1          53248(128K)   0.000  udp.proc
12336 8    10   Receive     1          53248(128K)   0.000  udp.proc
```

Table 132 describes the significant fields shown in the display.

Table 132 show processes (Software Modularity) Field Descriptions

Field	Description
PID	Process ID.
TID	Task ID.
Prio	Process priority.
STATE	Current state of process.
Blocked	Thread (with given process ID) that is currently blocked by the process.
Stack	Size, in kilobytes, of the memory stack.
CPU	CPU time, in minutes and seconds, used by the process.
Name	Process name.

Related Commands

Command	Description
show processes cpu	Displays detailed CPU utilization statistics (CPU use per process) when a Software Modularity image is running.
show processes detailed	Displays detailed information about POSIX and Cisco IOS processes when a Software Modularity image is running.
show processes kernel	Displays information about System Manager kernel processes when a Software Modularity image is running.
show processes memory	Displays amount of system memory used per system process.

show processes cpu

To display detailed CPU utilization statistics (CPU use per process) when Cisco IOS or Cisco IOS Software Modularity images are running, use the **show processes cpu** command in privileged EXEC mode.

Cisco IOS Software

show processes cpu [**history** | **sorted**]

Cisco IOS Software Modularity

show processes cpu [**detailed** [*process-id* | *process-name*] | **history**]

Syntax Description

history	(Optional) Displays CPU history in a graph format.
sorted	(Optional) For cisco IOS images only. Displays CPU utilization sorted by percentage.
detailed	(Optional) For Cisco IOS Software Modularity images only. Displays more detailed information about Cisco IOS processes (not for POSIX processes).
process-id	(Optional) For Cisco IOS Software Modularity images only. Process identifier.
process-name	(Optional) For Cisco IOS Software Modularity images only. Process name.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.0	This command was introduced.
12.2(2)T	The **history** keyword was added.
12.3(8)	This command was enhanced to display ARP output.
12.3(14)T	This command was enhanced to display ARP output.
12.2(18)SXF4	This command was enhanced to support Cisco IOS Software Modularity images.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

Cisco IOS Software

If you use the optional **history** keyword, three graphs are displayed for Cisco IOS images:

- CPU utilization for the last 60 seconds
- CPU utilization for the last 60 minutes
- CPU utilization for the last 72 hours

Maximum usage is measured and recorded every second; average usage is calculated on periods of more than one second. Consistently high CPU utilization over an extended period of time indicates a problem and using the **show processes cpu** command is useful for troubleshooting. Also, you can use the output of this command in the Cisco Output Interpreter tool to display potential issues and fixes. Output Interpreter is available to registered users of Cisco.com who are logged in and have Java Script enabled.

For a list of system processes, go to http://www.cisco.com/warp/public/63/showproc_cpu.html.

Cisco IOS Software Modularity

Cisco IOS Software Modularity images display only one graph that shows the CPU utilization for the last 60 minutes. The horizontal axis shows times (for example, 0, 5, 10, 15 minutes), and the vertical axis shows total percentage of CPU utilization (0 to 100 percent).

Examples

Example output varies between Cisco IOS software images and Cisco IOS Software Modularity software images. To view the appropriate output, choose one of the following sections:

- Cisco IOS Software
- Cisco IOS Software Modularity

Cisco IOS Software

The following is sample output from the **show processes cpu** command without keywords:

```
Router# show processes cpu

CPU utilization for five seconds: 5%/2%; one minute: 3%; five minutes: 2%
  PID  Runtime (ms)    Invoked   uSecs   5Sec  1Min  5Min  TTY  Process
    1          1736         58   29931     0%    0%    0%  0    Check heaps
    2            68        585     116  1.00% 1.00%    0%  0    IP Input
    3             0        744       0     0%    0%    0%  0    TCP Timer
    4             0          2       0     0%    0%    0%  0    TCP Protocols
    5             0          1       0     0%    0%    0%  0    BOOTP Server
    6            16        130     123     0%    0%    0%  0    ARP Input
    7             0          1       0     0%    0%    0%  0    Probe Input
    8             0          7       0     0%    0%    0%  0    MOP Protocols
    9             0          2       0     0%    0%    0%  0    Timers
   10           692         64   10812     0%    0%    0%  0    Net Background
   11             0          5       0     0%    0%    0%  0    Logger
   12             0         38       0     0%    0%    0%  0    BGP Open
   13             0          1       0     0%    0%    0%  0    Net Input
   14           540       3466     155     0%    0%    0%  0    TTY Background
   15             0          1       0     0%    0%    0%  0    BGP I/O
   16          5100       1367    3730     0%    0%    0%  0    IGRP Router
   17            88       4232      20  0.20% 1.00%    0%  0    BGP Router
   18           152      14650      10     0%    0%    0%  0    BGP Scanner
   19           224         99    2262     0%    0% 1.00%  0    Exec
```

The following is sample output of the one-hour portion of the output. The Y-axis of the graph is the CPU utilization. The X-axis of the graph is the increment within the time period displayed in the graph. This example shows the individual minutes during the previous hour. The most recent measurement is on the left of the X-axis.

```
Router# show processes cpu history

!--- One minute output omitted

6665776865756676676666667667677676766666766767767666566667
6378016198993513709771991443732358689932740858269643922613
100
```

```
90
80          *  *                     * *     *  * *  *
70  * * ***** *  ** ***** ***  **** ******  *  *******     * *
60  #***##*##*#***#####*#*#######*#*###*#*#*##*#*##*#*######*#
50  ############################################################
40  ############################################################
30  ############################################################
20  ############################################################
10  ############################################################
    0....5....1....1....2....2....3....3....4....4....5....5....
              0    5    0    5    0    5    0    5    0    5
               CPU% per minute (last 60 minutes)
              * = maximum CPU% # = average CPU%

!--- 72-hour output omitted
```

The top two rows, read vertically, display the highest percentage of CPU utilization recorded during the time increment. In this example, the CPU utilization for the last minute recorded is 66 percent. The device may have reached 66 percent only once during that minute, or it may have reached 66 percent multiple times. The device records only the peak reached during the time increment and the average over the course of that increment.

The following is sample output from the **show processes cpu** command that shows an ARP probe process:

```
Router# show processes cpu | include ARP

17       38140    389690          97  0.00%  0.00%  0.00%   0 ARP Input
36           0         1           0  0.00%  0.00%  0.00%   0 IP ARP Probe
40           0         1           0  0.00%  0.00%  0.00%   0 ATM ARP INPUT
80           0         1           0  0.00%  0.00%  0.00%   0 RARP Input
114          0         1           0  0.00%  0.00%  0.00%   0 FR ARP
```

Table 133 describes the fields shown in the output.

Table 133 show processes cpu Field Descriptions

Field	Description
CPU utilization for five seconds	CPU utilization for the last 5 seconds. The second number indicates the percent of CPU time spent at the interrupt level.
one minute	CPU utilization for the last minute.
five minutes	CPU utilization for the last 5 minutes.
PID	Process ID.
Runtime (ms)	CPU time that the process has used (in milliseconds).
Invoked	Number of times that the process has been invoked.
uSecs	Microseconds of CPU time for each process invocation.
5Sec	CPU utilization by task in the last 5 seconds.
1Min	CPU utilization by task in the last minute.
5Min	CPU utilization by task in the last 5 minutes.
TTY	Terminal that controls the process.
Process	Name of the process.

Note Because platforms have a 4- to 8-millisecond clock resolution, run times are considered reliable only after several invocations or a reasonable, measured run time.

Cisco IOS Software Modularity

The following is sample output from the **show processes cpu** command when a Software Modularity image is running:

```
Router# show processes cpu

Total CPU utilization for 5 seconds: 99.6%; 1 minute: 98.5%; 5 minutes: 85.3%
PID        5Sec    1Min     5Min Process
1          0.0%    0.1%     0.8% kernel
3          0.0%    0.0%     0.0% qdelogger
4          0.0%    0.0%     0.0% devc-pty
6          0.7%    0.2%     0.1% devc-ser2681
7          0.0%    0.0%     0.0% dumper.proc
4104       0.0%    0.0%     0.0% pipe
8201       0.0%    0.0%     0.0% mqueue
8202       0.0%    0.0%     0.0% fsdev.proc
8203       0.0%    0.0%     0.0% flashfs_hes_slot1.proc
8204       0.0%    0.0%     0.0% flashfs_hes_slot0.proc
8205       0.0%    0.0%     0.0% flashfs_hes_bootflash.proc
8206       0.0%    0.0%     0.0% dfs_disk2.proc
8207       0.0%    0.0%     0.0% dfs_disk1.proc
8208       0.0%    0.0%     0.0% dfs_disk0.proc
8209       0.0%    0.0%     0.0% ldcache.proc
8210       0.0%    0.0%     0.0% watchdog.proc
8211       0.0%    0.0%     0.0% syslogd.proc
8212       0.0%    0.0%     0.0% name_svr.proc
8213       0.0%    0.1%     0.0% wdsysmon.proc
8214       0.0%    0.0%     0.0% sysmgr.proc
8215       0.0%    0.0%     0.0% kosh.proc
12290      0.0%    0.0%     0.0% chkptd.proc
12312      0.0%    0.0%     0.0% sysmgr.proc
12313      0.0%    0.0%     0.0% syslog_dev.proc
12314      0.0%    0.0%     0.0% itrace_exec.proc
12315      0.0%    0.0%     0.0% packet.proc
12316      0.0%    0.0%     0.0% installer.proc
12317     29.1%   28.5%    19.6% ios-base
12318      0.0%    0.0%     0.0% fh_fd_oir.proc
12319      0.0%    0.0%     0.1% fh_fd_cli.proc
12320      0.0%    0.0%     0.0% fh_metric_dir.proc
12321      0.0%    0.0%     0.0% fh_fd_snmp.proc
12322      0.0%    0.0%     0.0% fh_fd_none.proc
12323      0.0%    0.0%     0.0% fh_fd_intf.proc
12324     48.5%   48.5%    35.8% iprouting.iosproc
12325      0.0%    0.0%     0.0% fh_fd_timer.proc
12326      0.0%    0.0%     0.0% fh_fd_ioswd.proc
12327      0.0%    0.0%     0.0% fh_fd_counter.proc
12328      0.0%    0.0%     0.0% fh_fd_rf.proc
12329      0.0%    0.0%     0.0% fh_server.proc
12330      0.0%    0.0%     0.0% cdp2.iosproc
12331      0.0%    0.0%     0.0% fh_policy_dir.proc
12332      0.0%    0.0%     0.0% ipfs_daemon.proc
12333      0.0%    0.0%     0.0% raw_ip.proc
12334      0.0%    0.0%     0.0% inetd.proc
12335     19.1%   20.4%    12.6% tcp.proc
12336      0.0%    0.0%     0.0% udp.proc
```

Table 134 describes the significant fields shown in the display.

Table 134 show processes cpu (Software Modularity) Field Descriptions

Field	Description
Total CPU utilization for five seconds	Total CPU utilization for the last 5 seconds. The second number indicates the percent of CPU time spent at the interrupt level.
one minute	Total CPU utilization for the last minute.
five minutes	Total CPU utilization for the last 5 minutes.
PID	Process ID.
5Sec	Percentage of CPU time spent at the interrupt level for this process during the last five seconds.
1Min	Percentage of CPU time spent at the interrupt level for this process during the last minute.
5Min	Percentage of CPU time spent at the interrupt level for this process during the last five minutes.
Process	Process name.

The following is partial sample output from the **show processes cpu** command with the **detailed** keyword when a Software Modularity image is running:

```
Router# show processes cpu detailed

Total CPU utilization for 5 seconds: 99.6%; 1 minute: 99.3%; 5 minutes: 88.6%
PID/TID   5Sec    1Min     5Min Process              Prio  STATE        CPU
1         0.0%    0.7%     0.7% kernel                                   8.900
      1   0.4%    0.7%    11.4% [idle thread]          0  Ready        2m28s
      2   0.0%    0.0%     0.0%                       63  Receive      0.000
      3   0.0%    0.0%     0.0%                       10  Receive      0.000
      4   0.0%    0.0%     0.1%                       11  Receive      1.848
      5   0.0%    0.0%     0.0%                       63  Receive      0.000
.
.
.
PID/TID   5Sec    1Min     5Min Process              Prio  STATE        CPU
8214      0.0%    0.0%     0.0% sysmgr.proc                              0.216
      1   0.0%    0.0%     0.0%                       10  Receive      0.132
      2   0.0%    0.0%     0.0%                       10  Sigwaitin    0.000
      3   0.0%    0.0%     0.0%                       10  Receive      0.004
      4   0.0%    0.0%     0.0%                       10  Receive      0.000
      5   0.0%    0.0%     0.0%                       10  Receive      0.000
      6   0.0%    0.0%     0.0%                       10  Receive      0.004
      7   0.0%    0.0%     0.0%                       10  Receive      0.000
      8   0.0%    0.0%     0.0%                       10  Receive      0.000
      9   0.0%    0.0%     0.0%                       10  Receive      0.000
     10   0.0%    0.0%     0.0%                       10  Receive      0.000
     11   0.0%    0.0%     0.0%                       10  Receive      0.000
     12   0.0%    0.0%     0.0%                       10  Receive      0.000
     13   0.0%    0.0%     0.0%                       10  Receive      0.028
     14   0.0%    0.0%     0.0%                       10  Receive      0.040
     15   0.0%    0.0%     0.0%                       10  Receive      0.000
     16   0.0%    0.0%     0.0%                       10  Receive      0.000
     17   0.0%    0.0%     0.0%                       10  Receive      0.004
     18   0.0%    0.0%     0.0%                       10  Receive      0.000
     19   0.0%    0.0%     0.0%                       10  Receive      0.000
     20   0.0%    0.0%     0.0%                       10  Receive      0.000
```

```
     21   0.0%    0.0%     0.0%                      10  Receive      0.004
     22   0.0%    0.0%     0.0%                      10  Receive      0.000
PID/TID   5Sec    1Min     5Min Process             Prio  STATE        CPU
8215      0.0%    0.0%     0.0% kosh.proc                              0.044
      1   0.0%    0.0%     0.0%                      10  Reply        0.044
PID/TID   5Sec    1Min     5Min Process             Prio  STATE        CPU
12290     0.0%    0.0%     0.0% chkptd.proc                            0.080
      1   0.0%    0.0%     0.0%                      10  Receive      0.080
      2   0.0%    0.0%     0.0%                      10  Receive      0.000
PID/TID   5Sec    1Min     5Min Process             Prio  STATE        CPU
12312     0.0%    0.0%     0.0% sysmgr.proc                            0.112
      1   0.0%    0.0%     0.0%                      10  Receive      0.112
      2   0.0%    0.0%     0.0%                      10  Sigwaitin    0.000
PID/TID   5Sec    1Min     5Min Process             Prio  STATE        CPU
12316     0.0%    0.0%     0.0% installer.proc                         0.072
      1   0.0%    0.0%     0.0%                      10  Receive      0.000
      3   0.0%    0.0%     0.0%                      10  Nanosleep    0.000
      4   0.0%    0.0%     0.0%                      10  Sigwaitin    0.000
      6   0.0%    0.0%     0.0%                      10  Receive      0.000
Process sbin/ios-base, type IOS, PID = 12317
CPU utilization for five seconds: 12%/9%; one minute: 13%; five minutes: 10%
Task  Runtime(ms)  Invoked  uSecs    5Sec   1Min   5Min TTY Task Name
   1         219     1503    145   0.00%  0.00%  0.00%   0 Hot Service Task
   2       23680    42384    558   2.39%  6.72%  4.81%   0 Service Task
   3        6104    11902    512   3.51%  1.99%  1.23%   0 Service Task
   4        1720     5761    298   1.91%  0.90%  0.39%   0 Service Task
   5           0        5      0   0.00%  0.00%  0.00%   0 Chunk Manager
   6           0        1      0   0.00%  0.00%  0.00%   0 Connection Mgr
   7           4      106     37   0.00%  0.00%  0.00%   0 Load Meter
   8        6240     7376    845   0.23%  0.15%  0.55%   0 Exec
   9         379       62   6112   0.00%  0.07%  0.04%   0 Check heaps
  10           0        1      0   0.00%  0.00%  0.00%   0 Pool Manager
  11           3        2   1500   0.00%  0.00%  0.00%   0 Timers
  12           0        1      0   0.00%  0.00%  0.00%   0 AAA_SERVER_DEADT
  13           0        2      0   0.00%  0.00%  0.00%   0 AAA high-capacit
  14         307      517    593   0.00%  0.05%  0.03%   0 EnvMon
  15           0        1      0   0.00%  0.00%  0.00%   0 OIR Handler
  16         283       58   4879   0.00%  0.04%  0.02%   0 ARP Input
  17           0        2      0   0.00%  0.00%  0.00%   0 Serial Backgroun
  18           0       81      0   0.00%  0.00%  0.00%   0 ALARM_TRIGGER_SC
  19           0        2      0   0.00%  0.00%  0.00%   0 DDR Timers
  20           0        2      0   0.00%  0.00%  0.00%   0 Dialer event
  21           4        2   2000   0.00%  0.00%  0.00%   0 Entity MIB API
  22           0       54      0   0.00%  0.00%  0.00%   0 Compute SRP rate
  23           0        9      0   0.00%  0.00%  0.00%   0 IPC Dynamic Cach
  24           0        1      0   0.00%  0.00%  0.00%   0 IPC Zone Manager
  25           0        1      0   0.00%  0.00%  0.00%   0 IPC Punt Process
  26           4      513      7   0.00%  0.00%  0.00%   0 IPC Periodic Tim
  27          11      513     21   0.00%  0.00%  0.00%   0 IPC Deferred Por
  28           0        1      0   0.00%  0.00%  0.00%   0 IPC Seat Manager
  29          83     1464     56   0.00%  0.00%  0.00%   0 EEM ED Syslog
.
.
.
```

Table 135 describes the significant fields shown in the display.

Table 135 show processes cpu detailed (Software Modularity) Field Descriptions

Field	Description
Total CPU utilization for five seconds	Total CPU utilization for the last 5 seconds. The second number indicates the percent of CPU time spent at the interrupt level.
one minute	Total CPU utilization for the last minute.
five minutes	Total CPU utilization for the last 5 minutes.
PID/TID	Process ID or task ID.
5Sec	Percentage of CPU time spent at the interrupt level for this process during the last five seconds.
1Min	Percentage of CPU time spent at the interrupt level for this process during the last minute.
5Min	Percentage of CPU time spent at the interrupt level for this process during the last five minutes.
Process	Process name.
Prio	Priority level of the process.
STATE	Current state of the process.
CPU	CPU utilization of the process in minutes and seconds.
type	Type of process; can be either IOS or POSIX.
Task	Task sequence number.
Runtime(ms)	CPU time that the process has used (in milliseconds).
Invoked	Number of times that the process has been invoked.
uSecs	Microseconds of CPU time for each process invocation.
5Sec	CPU utilization by task in the last 5 seconds.
1Min	CPU utilization by task in the last minute.
5Min	CPU utilization by task in the last 5 minutes.
TTY	Terminal that controls the process.
Task Name	Task name.

Related Commands

Command	Description
show processes	Displays information about active processes.
show processes memory	Displays the amount of system memory used per system process.

show processes interrupt mask buffer

To display information in the interrupt mask buffer, use the **show processes interrupt mask buffer** command in privileged EXEC mode.

show processes interrupt mask buffer

buffer	Displays stack trace and information about the places where interrupts have been masked more than the configured threshold time.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.4(2)T	This command was introduced.

Examples

The following is sample output from the **show processes interrupt mask buffer** command. The output displays stack trace and relevant information about the places where interrupts have been masked more than the configured threshold time:

```
Router# show processes interrupt mask buffer

Allowable interrupt mask time : 50 micro seconds
 Allowable number of half pipeline ticks for this platform : 5000
 Buffer Size            : 50 entries
 NETS Disable       : 3
 TTY  Disable        : 4
 ALL  Disable        : 4
 emt_call          : 11
 disable_interrupts : 12

  PID  Level  Time Spent(us)  Count  Stack Trace

    3   11      360            1       0x608C3C14 0x60894748 0x6089437C 0x608943AC
0x609CEC88 0x609CECFC 0x609C8524
    3   11      322            1       0x608C3C14 0x608943BC 0x609CEC88 0x609CECFC
0x609C8524 0x60867C28 0x607C70B0
    3   4       147            1       0x6078AED4 0x6078BE94 0x6078C750 0x6078C8D4
0x607E27F0 0x607E27C0 0x607E50B0
```

Related Commands

Command	Description
clear processes interrupt mask detail	Clears the interrupt masked details for all processes and stack traces which have been dumped into the interrupt mask buffer.
scheduler interrupt mask profile	Enables or disables interrupt mask profiling for all processes running on the system.
scheduler interrupt mask size	Configures the maximum number of entries that can exist in the interrupt mask buffer.

Command	Description
scheduler interrupt mask time	Configures the maximum amount of time a process can run with interrupts masked.
show processes interrupt mask detail	Displays interrupt masked details for the specified process or all processes in the system.

show processes interrupt mask detail

To display information about interrupt masking, use the **show processes interrupt mask detail** command in privileged EXEC mode.

show processes interrupt mask detail [*pid]*

Syntax Description

detail	Displays information about the total amount of time and the number of times interrupts have been masked by all processes.
pid	(Optional) An integer that specifies the process id for which to display the total accumulated time and the number of times interrupts have been masked.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.4(2)T	This command was introduced.

Examples

The following is sample output from the **show processes interrupt mask detail** command. the output displays information about the total amount of time and number of times interrupts have been masked by all processes:

```
Router# show processes interrupt mask detail

PID  Time Spent(us)    Count  Process Name

2       6388           1791   Load Meter
3       7957           16831  Exec
5       6710           2813   Check heaps
```

The following is sample output from the show processes interrupt mask detail command with the process ID specified. The output displays the total time (accumulative), number of times interrupts have been masked by a specific process:

```
Router# show processes interrupt mask detail 2

Process ID      : 2
 Process Name : Load Meter
 Total Interrupt Masked Time  : 6586 (us)
 Total Interrupt Masked Count : 1845
```

Related Commands

Command	Description
clear processes interrupt mask detail	Clears the interrupt masked details for all processes and stack traces which have been dumped into the interrupt mask buffer.
scheduler interrupt mask profile	Enables or disables interrupt mask profiling for all processes running on the system.

Command	Description
scheduler interrupt mask size	Configures the maximum number of entries that can exist in the interrupt mask buffer.
scheduler interrupt mask time	Configures the maximum amount of time a process can run with interrupts masked.
show processes interrupt mask buffer	Displays the information stored in the interrupt mask buffer.

show processes memory

To show the amount of memory used by each system process in Cisco IOS or Cisco IOS Software Modularity images, use the **show processes memory** command in privileged EXEC mode.

Cisco IOS Software

show processes memory [*process-id* | **sorted** [**allocated** | **getbufs** | **holding**]]

Cisco IOS Software Modularity

show processes memory [**detailed** [*process-name*[**:***instance-id*] | *process-id* [**taskid** *task-id*]]] [**alloc-summary** | **sorted** {**start** | **size** | **caller**}]

Syntax Description

Cisco IOS Software Syntax	
process-id	(Optional) Process ID (PID) of a specific process. When you specify a process ID, only details for the specified process will be shown.
sorted	(Optional) Displays memory data sorted by the "Allocated, " "Getbufs," or "Holding" column. If the **sorted** keyword is used by itself, data is sorted by the "Holding" column by default.
allocated	(Optional) Displays memory data sorted by the "Allocated" column.
getbufs	(Optional) Displays memory data sorted by the "Getbufs" (Get Buffers) column.
holding	(Optional) Displays memory data sorted by the "Holding" column. This is the default.
Cisco IOS Software Modularity Syntax	
detailed	(Optional) Displays detailed information about iosproc processes.
process-name	(Optional) Process name.
:*instance-id*	(Optional) Instance name of either the Cisco IOS task or POSIX process. The colon is required.
process-id	(Optional) Process identifier.
taskid	(Optional) Displays detailed memory usage of a Cisco IOS task within a process.
task-id	(Optional) Cisco IOS task identifier.
alloc-summary	(Optional) Displays summary POSIX process memory usage per allocator.
sorted	(Optional) Displays POSIX process memory usage sorted by start address, size, or the PC that called the process.
start	(Optional) Displays POSIX process memory usage sorted by start address of the process.
size	(Optional) Displays POSIX process memory usage sorted by size of the process.
caller	(Optional) Displays POSIX process memory usage sorted by the PC that called the process.

Command Default

Cisco IOS Software

The memory used by all types of system processes is displayed.

Cisco IOS Software Modularity

The system memory followed by a one-line summary of memory information about each Software Modularity process is displayed.

Command Modes

Privileged Exec (#)

Command History

Release	Modification
10.0	This command was introduced.
12.0(23)S	The **sorted** [**allocated** | **getbufs** | **holding**] syntax was introduced. [CSCdy22469]
12.2(13)	The **sorted** [**allocated** | **getbufs** | **holding**] syntax was integrated in Cisco IOS Release 12.2(13).
12.2(13)S	The **sorted** [**allocated** | **getbufs** | **holding**] syntax was integrated in Cisco IOS Release 12.2(13)S.
12.2(13)T	The **sorted** [**allocated** | **getbufs** | **holding**] syntax was integrated in Cisco IOS Release 12.2(13)T.
12.0(28)S	The output of the header line was updated to support the Memory Thresholding feature.
12.2(22)S	The output of the header line was updated to support the Memory Thresholding feature.
12.3(7)T	The output of the header line was updated to support the Memory Thresholding feature.
12.0(30)S	The summary information (first lines of output) for this command was separated out and labeled by memory pool type (Total Process Memory, Total I/O Memory, and so on). This enhancement also corrected a total process memory mismatch error (mismatch between **show processes memory**, **show processes memory sorted**, and **show memory** and its variants).
12.2(28)S	The summary information (first lines of output) for this command was separated out and labeled by memory pool type (Total Process Memory, Total I/O Memory, and so on). This enhancement also corrected a total process memory mismatch error (mismatch between **show processes memory**, **show processes memory sorted**, and **show memory** and its variants).
12.3(11)T	The summary information (first lines of output) for this command was separated out and labeled by memory pool type (Total Process Memory, Total I/O Memory, and so on). This enhancement also corrected a total process memory mismatch error (mismatch between **show processes memory**, **show processes memory sorted**, and **show memory** and its variants).
12.2(18)SXF4	The syntax was modified to support Cisco IOS Software Modularity images.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

The **show processes memory** command (and **show processes memory sorted** command) displays a summary of total, used, and free memory, followed by a list of processes and their memory impact.

If the standard **show processes memory** *process-id* command is used, processes are sorted by their process ID (PID). If the **show processes memory sorted** command is used, the default sorting is by the Holding value.

Output Prior to Releases 12.3(7)T, 12.2(22)S, and 12.0(28)S

The first line (header line) of the **show processes memory** [**sorted**] command listed Total memory, Used memory, and Free memory values.

Output in Releases 12.3(7)T, 12.3(8)T, 12.2(22)S Through 12.2(27)S2, 12.0(28)S, and 12.0(29)S

In Releases 12.3(7)T, 12.2(22)S, and 12.0(28)S, the "Memory Thresholding" feature was introduced. This feature affected the header line and the "Holding" column of the **show processes memory** command as follows.

The value for "Total" in the **show processes memory** command and the values listed in the "Holding" column, showed the total (cumulative) value for the processor memory pools and the alternate memory pool* (typically, the I/O memory pool). However, the **show processes memory sorted** version of this command, and other commands, such as the **show memory summary** command, did not include the alternate memory pool in the totals (in other words, these commands showed the total value for the Processor memory pool only). This caused an observed mismatch of memory totals between commands.

If you are using these releases, use the output of **show memory summary** command to determine the individual amounts of Total and Free memory for the Processor memory pool and the I/O memory pool.

Output in Releases 12.3(11)T, 12.2(28)S, 12.0(30)S and Later Releases

Beginning in Releases 12.3(11)T, 12.2(28)S, and 12.0(30)S, the summary information (first output lines) for the **show processes memory** command is separated by memory pool. For example, there are now individual lines for "Total Process Memory," "Total I/O Memory," and "Total PCI Memory." If using these releases or later releases, your Total Process Memory should match the total process memory shown for other commands, such as the **show memory summary** command.

About Alternate Memory Pools

An "alternate memory pool" is a memory pool which can be used as an alternative to allocate memory when the target (main) memory pool has been filled. For example, many platforms have a memory type called "Fast" that is limited to a small size (because the memory media used for Fast memory is expensive). To prevent memory allocations from failing once the available Fast memory has been used up, the normal Processor memory can be configured as an alternative memory pool for the Fast memory pool.

Cisco IOS Software Modularity

Use the **show processes memory** command without any arguments and keywords to display the system memory followed by a one-line summary of memory information about each modular Cisco IOS process. Use the **detailed** keyword with this command to display detailed memory information about all processes. Other arguments and keywords are used to display Cisco IOS Software Modularity process memory information for a specified process name or process ID.

Examples

Example output varies between Cisco IOS software releases. To view the appropriate output, choose one of the following sections:

- show processes memory Command for Releases Prior to 12.3(11)T, 12.2(28)S, and 12.0(30)S
- show processes memory Command for Cisco IOS Software Modularity

show processes memory Command for Releases Prior to 12.3(7)T, 12.2(22)S, and 12.0(28)S

The following is sample output from the **show processes memory** command:

```
Router# show processes memory

Processor Pool Total:   25954228 Used:    8368640 Free:   17585588

 PID TTY  Allocated      Freed    Holding    Getbufs    Retbufs Process
   0   0    8629528     689900    6751716          0          0 *Init*
   0   0      24048      12928      24048          0          0 *Sched*
   0   0        260        328         68     350080          0 *Dead*
   1   0          0          0      12928          0          0 Chunk Manager
   2   0        192        192       6928          0          0 Load Meter
   3   0     214664        304     227288          0          0 Exec
   4   0          0          0      12928          0          0 Check heaps
   5   0          0          0      12928          0          0 Pool Manager
   6   0        192        192      12928          0          0 Timers
   7   0        192        192      12928          0          0 Serial Backgroun
   8   0        192        192      12928          0          0 AAA high-capacit
   9   0          0          0      24928          0          0 Policy Manager
  10   0          0          0      12928          0          0 ARP Input
  11   0        192        192      12928          0          0 DDR Timers
  12   0          0          0      12928          0          0 Entity MIB API
  13   0          0          0      12928          0          0 MPLS HC Counter
  14   0          0          0      12928          0          0 SERIAL A'detect
.
.
.
  78   0          0          0      12992          0          0 DHCPD Timer
  79   0        160          0      13088          0          0 DHCPD Database
                                  8329440 Total
```

Table 136 describes the significant fields shown in the display.

Table 136 show processes memory Field Descriptions

Field	Description
Processor Pool Total	Total amount of memory, in kilobytes, held for the Processor memory pool.
Used	Total amount of used memory, in kilobytes, in the Processor memory pool.
Free	Total amount of free memory, in kilobytes, in the Processor memory pool.
PID	Process ID.
TTY	Terminal that controls the process.
Allocated	Bytes of memory allocated by the process.
Freed	Bytes of memory freed by the process, regardless of who originally allocated it.
Holding	Amount of memory, in kilobytes, currently allocated to the process.
Getbufs	Number of times the process has requested a packet buffer.
Retbufs	Number of times the process has relinquished a packet buffer.
Process	Process name.
Init	System initialization process.

Table 136 show processes memory Field Descriptions (continued)

Field	Description
Sched	The scheduler process.
Dead	Processes as a group that are now dead.
<value> Total	Total amount of memory, in kilobytes, held by all processes (sum of the "Holding" column).

The following is sample output from the **show processes memory** command when the **sorted** keyword is used. In this case, the output is sorted by the "Holding" column, from largest to smallest.

```
Router# show processes memory sorted

Processor Pool Total:   25954228 Used:    8371280 Free:   17582948

 PID TTY  Allocated      Freed    Holding    Getbufs    Retbufs Process
   0   0    8629528     689900    6751716          0          0 *Init*
   3   0     217304        304     229928          0          0 Exec
  53   0     109248        192      96064          0          0 DHCPD Receive
  56   0          0          0      32928          0          0 COPS
  19   0      39048          0      25192          0          0 Net Background
  42   0          0          0      24960          0          0 L2X Data Daemon
  58   0        192        192      24928          0          0 X.25 Background
  43   0        192        192      24928          0          0 PPP IP Route
  49   0          0          0      24928          0          0 TCP Protocols
  48   0          0          0      24928          0          0 TCP Timer
  17   0        192        192      24928          0          0 XML Proxy Client
   9   0          0          0      24928          0          0 Policy Manager
  40   0          0          0      24928          0          0 L2X SSS manager
  29   0          0          0      24928          0          0 IP Input
  44   0        192        192      24928          0          0 PPP IPCP
  32   0        192        192      24928          0          0 PPP Hooks
  34   0          0          0      24928          0          0 SSS Manager
  41   0        192        192      24928          0          0 L2TP mgmt daemon
  16   0        192        192      24928          0          0 Dialer event
  35   0          0          0      24928          0          0 SSS Test Client
 --More--
```

The following is sample output from the **show processes memory** command when a Process ID (*process-id*) is specified:

```
Router# show processes memory 1

Process ID: 1
Process Name: Chunk Manager
Total Memory Held: 8428 bytes

Processor memory holding = 8428 bytes
pc = 0x60790654, size =      6044, count =    1
pc = 0x607A5084, size =      1544, count =    1
pc = 0x6076DBC4, size =       652, count =    1
pc = 0x6076FF18, size =       188, count =    1

I/O memory holding = 0 bytes

Router# show processes memory 2

Process ID: 2
Process Name: Load Meter
Total Memory Held: 3884 bytes
```

```
Processor memory holding = 3884 bytes
pc = 0x60790654, size =      3044, count =    1
pc = 0x6076DBC4, size =       652, count =    1
pc = 0x6076FF18, size =       188, count =    1

I/O memory holding = 0 bytes
```

show processes memory Command for Releases Prior to 12.3(11)T, 12.2(28)S, and 12.0(30)S

The following example shows the output of the **show processes memory** command before the changes to the summary information were made. Note that the "Total:" in the **show processes summary** command indicates total memory for all memory pools; in this example, the **show processes memory** Total of 35423840 can be obtained by adding the Processor and I/O totals shown in the output of the **show memory summary** command. Note also that the **show processes memory sorted** command lists the Total Processor Memory (matches the **show memory summary** Processor Total, but the **show processes memory** command (without the **sorted** keyword) lists the Total for all memory pools (Processor plus I/O memory).

```
Router# show version | include IOS

Cisco IOS Software, 3600 Software (C3660-BIN-M), Version 12.3(9)

Router# show memory summary

               Head    Total(b)     Used(b)     Free(b)   Lowest(b)  Largest(b)
Processor   61E379A0    27035232     8089056    18946176    17964108    17963664
      I/O    3800000     8388608     2815088     5573520     5561520     5573472

.
.
.

Router# show processes memory

Total: 35423840, Used: 10904192, Free: 24519648
 PID TTY  Allocated      Freed    Holding    Getbufs    Retbufs Process
   0   0   14548868    3004980    9946092          0          0 *Init*
   0   0      12732     567448      12732          0          0 *Sched*
.
.
.

Router# show processes memory sorted

Total: 27035232, Used: 8089188, Free: 18946044
 PID TTY  Allocated      Freed    Holding    Getbufs    Retbufs Process
   0   0   14548868    3004980    9946092          0          0 *Init*
  64   0      76436       3084      74768          0          0 CEF process
.
.
.

Router# show version | include IOS

Cisco IOS Software, 3600 Software (c3660-p-mz), Version 12.0(29)S,

Router# show memory summary

               Head      Total(b)     Used(b)     Free(b)   Lowest(b)  Largest(b)
Processor    126CB10    49,331,668     6454676    42876992    42642208    42490796

Router# show processes memory
```

```
Total: 50,994,868, Used: 6220092, Free: 44774776
 PID TTY  Allocated      Freed    Holding    Getbufs    Retbufs Process
   0   0    6796228     627336    5325956          0          0 *Init*
   0   0        200      29792        200          0          0 *Sched*
   0   0        192        744          0     349000          0 *Dead*
   1   0          0          0      12896          0          0 Chunk Manager
.
.

Router# show processes memory sorted

Total: 50,994,868, Used: 6222644, Free: 44772224
 PID TTY  Allocated      Freed    Holding    Getbufs    Retbufs Process
   0   0    6796228     627336    5325956          0          0 *Init*
  13   0      39056          0      25264          0          0 Net Background
  48   0          0          0      24896          0          0 L2X SSS manager
  18   0          0          0      24896          0          0 IP Input
.
.
```

show processes memory Command for Cisco IOS Software Modularity

The following is sample output from the **show processes memory** command when a Cisco IOS Software Modularity image is running:

```
Router# show processes memory

System Memory : 262144K total, 113672K used, 148472K free

PID       Text    Data   Stack  Dynamic     Total Process
1            0       0      12        0        12 kernel
12290       52       8      28      196       284 dumper.proc
3           12       8       8      144       172 devc-pty
4          132       8       8       32       180 devc-ser2681
6           16      12      24       48       100 pipe
8199        12      12       8       48        80 mqueue
8200        16      24      48      452       540 fsdev.proc
8201        52      20       8       96       176 flashfs_hes_slot1.proc
8202        52      20       8       80       160 flashfs_hes_bootflash.proc
8203        52      20       8      128       208 flashfs_hes_slot0.proc
8204        20      68      12      164       264 dfs_disk1.proc
8205        20      68      12      164       264 dfs_disk0.proc
8206        36       4       8      144       192 ldcache.proc
8207        32       8      20      164       224 syslogd.proc
8208        24       4      28      464       520 name_svr.proc
8209       124     104      28      344       600 wdsysmon.proc
8210       100     144      52      328       624 sysmgr.proc
8211        12       4      28       64       108 kosh.proc
12308      100     144      16      144       404 sysmgr.proc
12309       24       4      12      112       152 chkptd.proc
12310       12       4       8       96       120 syslog_dev.proc
12311       44       4      24      248       320 fh_metric_dir.proc
12312       36       4      24      216       280 fh_fd_snmp.proc
12313       36       4      24      216       280 fh_fd_intf.proc
12314       32       4      24      216       276 fh_fd_timer.proc
12315       40       4      24      216       284 fh_fd_ioswd.proc
12316       28       4      24      200       256 fh_fd_counter.proc
12317       80      20      44      368       512 fh_server.proc
12326      140      40      28      280       488 tcp.proc
12327       48       4      24      256       332 udp.proc
12328        4       4      28     4660      4696 iprouting.iosproc
12329        4       4      36      600       644 cdp2.iosproc
```

Table 137 describes the significant fields shown in the display.

Table 137 show processes memory (Software Modularity) Field Descriptions

Field	Description
total	Total amount of memory, in kilobytes, on the device.
used	Amount of memory, in kilobytes, used in the system.
free	Amount of free memory, in kilobytes, available in the system.
PID	Process ID.
Text	Amount of memory, in kilobytes, used by the text segment of the specified process.
Data	Amount of memory, in kilobytes, used by the data segment of the specified process.
Stack	Amount of memory, in kilobytes, used by the stack segment of the specified process.
Dynamic	Amount of memory, in kilobytes, used by the dynamic segment of the specified process.
Total	Total amount of memory, in kilobytes, used by the specified process.
Process	Process name.

The following is sample output from the **show processes memory** command with details about the memory of the process named cdp2.iosproc:

```
Router# show processes memory detailed cdp2.iosproc

System Memory : 262144K total, 113460K used, 148684K free

Process sbin/cdp2.iosproc, type IOS, PID = 12329
    640K total, 4K text, 4K data, 32K stack, 600K dynamic

Memory Summary for TaskID = 1
Holding = 10032

        PC       Size  Count
0x7322FC74       9192      1
0x73236538        640      1
0x73231E8C        200      1
```

The following is sample output from the **show processes memory** command with details about the memory of process 12322 and the task with the ID of 1:

```
Router# show processes memory detailed 12322 taskid 1

System Memory : 262144K total, 113456K used, 148688K free

Process sbin/c7200-p-blob, type IOS, PID = 12322
    16568K total, 16K text, 8K data, 64K stack, 16480K dynamic

Memory Summary for TaskID = 1
Holding = 10248

        PC       Size  Count
0x7322FC74       9192      1
0x73236538        640      1
0x73231E8C        256      1
```

```
0x74175060        160      1
```

Table 138 describes the significant fields shown in the display that are different from Table 137 on page 855.

Table 138 show processes memory detailed process-id taskid Field Descriptions

Field	Description
type	Type of process: POSIX or Cisco IOS.
Memory summary for TaskID	Task ID.
Holding	Amount of memory, in bytes, currently held by the task.
PC	Caller PC of the task.
Size	Amount of memory, in bytes, used by this task.
Count	Number of times that task has been called.

The following is sample output from the **show processes memory** command with details about the memory of POSIX process ID 234567 with summary process memory usage per allocator:

```
Router# show processes memory detailed 234567 alloc-summary

System Memory : 262144K total, 113672K used, 148472K free

Process sbin/sysmgr.proc, type POSIX, PID = 12308
    404K total, 100K text, 144K data, 16K stack, 144K dynamic
    81920 heapsize, 68620 allocated, 8896 free

Allocated Blocks
Address    Usize      Size       Caller
0x0806C358 0x00000478 0x000004D0 0x721C7290
0x0806D1E0 0x00000128 0x00000130 0x72B90248
0x0806D318 0x00003678 0x000036E0 0x72B9820C
0x0806D700 0x000002A0 0x000002C0 0x72B8EB58
0x0806D770 0x00000058 0x00000060 0x72BA5488
0x0806D7D8 0x000000A0 0x000000B0 0x72B8D228
0x0806D8A8 0x00000200 0x00000208 0x721A728C
0x0806FF78 0x00000068 0x00000070 0x72BA78EC
0x08071438 0x0000005C 0x00000068 0x72B908A8
0x08071508 0x0000010E 0x00000120 0x72BA7AFC
0x08072840 0x000000A8 0x000000C0 0x7270A060
0x08072910 0x0000010C 0x00000118 0x7273A898
0x08072A30 0x000000E4 0x000000F0 0x72749074
0x08072B28 0x000000B0 0x000000B8 0x7276E87C
0x08072BE8 0x0000006C 0x00000078 0x727367A4
0x08072C68 0x000000B8 0x000000C0 0x7271E2A4
0x08072D30 0x000000D0 0x000000D8 0x7273834C
0x08072E10 0x00000250 0x00000258 0x72718A70
0x08073070 0x000002F4 0x00000300 0x72726484
0x08073378 0x000006A8 0x000006B0 0x73EA4DC4
0x08073A30 0x00000060 0x00000068 0x7352A9F8
0x08073B38 0x00000068 0x00000070 0x72B92008
0x08073BB0 0x00000058 0x00000060 0x72B9201C
0x08073EB8 0x00002FB4 0x000031C0 0x08026FEC
0x08074028 0x000020B8 0x000020C0 0x72709C9C
0x08077400 0x000000A0 0x000000A8 0x721DED94
0x08078028 0x000022B8 0x000022C0 0x727446B8
0x0807C028 0x00002320 0x00002328 0x72B907C4

Free Blocks
```

```
Address    Size
0x0806FFF0 0x00000010
0x080714A8 0x00000058
0x08073E18 0x00000098
0x08073FE8 0x00000018
0x08076FA0 0x00000328
0x080774B0 0x00000B50
0x0807FFB8 0x00000048
0x08080028 0x00003FD8
```

Table 139 describes the significant fields shown in the display.

Table 139 show processes memory detailed alloc-summary Field Descriptions

Field	Description
heapsize	Size of the process heap, in kilobytes.
allocated	Amount of memory, in kilobytes, allocated from the heap.
free	Amount of free memory, in kilobytes, in the heap for the specified process.
Address	Block address, in hexadecimal.
Usize	Block size, in hexadecimal, without the trailer header.
Size	Block size, in hexadecimal.
Caller	Caller PC of the allocator of this block.

Related Commands

Command	Description
show memory	Displays statistics about memory, including memory-free pool statistics.
show processes	Displays information about the active processes.

show protocols

To display the configured protocols, use the **show protocols** command in EXEC mode.

This command shows the global and interface-specific status of any configured Level 3 protocol; for example, IP, DECnet, IPX, AppleTalk, and so on.

show protocols

Syntax Description

This command has no arguments or keywords.

Command Modes

EXEC

Command History

Release	Modification
10.0	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Examples

The following is sample output from the **show protocols** command:

```
Router# show protocols

Global values:
  Internet Protocol routing is enabled
  DECNET routing is enabled
  XNS routing is enabled
  Appletalk routing is enabled
  X.25 routing is enabled
Ethernet 0 is up, line protocol is up
  Internet address is 192.168.1.1, subnet mask is 255.255.255.0
  Decnet cost is 5
  XNS address is 2001.AA00.0400.06CC
  AppleTalk address is 4.129, zone Twilight
Serial 0 is up, line protocol is up
  Internet address is 192.168.7.49, subnet mask is 255.255.255.240
Ethernet 1 is up, line protocol is up
  Internet address is 192.168.2.1, subnet mask is 255.255.255.0
  Decnet cost is 5
  XNS address is 2002.AA00.0400.06CC
  AppleTalk address is 254.132, zone Twilight
Serial 1 is down, line protocol is down
  Internet address is 192.168.7.177, subnet mask is 255.255.255.240
  AppleTalk address is 999.1, zone Magnolia Estates
```

For more information on the parameters or protocols shown in this sample output, see the *Cisco IOS Network Protocols Configuration Guide, Part 1, Network Protocols Configuration Guide, Part 2*, and *Network Protocols Configuration Guide, Part 3*.

show region

To display valid memory regions (memory mapping) in use on your system, use the **show region** command in Privileged EXEC mode.

show region [**address** *hex-address*]

Syntax Description

address *hex-address*	(Optional) If a hex address is specified, this command will search the region list for the specified address.

Defaults

All memory regions are displayed.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1, 12.0(9)S	The **show region** command output was made available in the output of the **show technical-support** command.
12.2(15)ZN, 12.2(15)BZ, 12.1(14)E, 12.2(13)S, 12.2(13), 12.2(13)T, 12.0(23)S	The **show region** command was enabled as a separate command.
12.2(25)S, 12.3(14)T	The **show region** command output was updated to display information about free regions.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

This command can be useful for troubleshooting system bus errors. The system encounters a bus error when the processor tries to access a memory location that either does not exist (a software error) or does not respond properly (a hardware problem).

To use the **show region** command to troubleshoot a bus error, note the memory location address from the **show version** command, the **show context** command, or from the system error message that alerted you to the bus error. The **show region** command can then be used to determine if that address is a valid memory location.

For example, in the output of the **show version** command after a system restart casued by a bus error, you will see output similar to "System restarted by bus error at PC 0x30EE546, address 0xBB4C4." In this case, the memory location that the router tried to access is 0xBB4C4. If the address falls within one of the ranges in the **show region** output, it means that the router was accessing a valid memory address, but the hardware corresponding to that address is not responding properly. This indicates a hardware problem.

If the address reported by the bus error does not fall within the ranges displayed in the **show region** output, this means that the router was trying to access an address that is not valid. This indicates that it is a Cisco IOS software problem.

More detailed information is available on Cisco.com in Tech Note #7949, *Troubleshooting Bus Error Crashes*.

Examples

The following is sample output from the **show region** command:

```
Router# show region

Region Manager:

      Start         End      Size(b)  Class  Media  Name
 0x40000000  0x40001FFF         8192  Iomem  REG    qa
 0x40002000  0x401FFFFF      2088960  Iomem  R/W    memd
 0x48000000  0x48001FFF         8192  Iomem  REG    qa:writethru
 0x50002000  0x501FFFFF      2088960  Iomem  R/W    memd:(memd_bitswap)
 0x58002000  0x581FFFFF      2088960  Iomem  R/W    memd:(memd_uncached)
 0x60000000  0x6FFFFFFF    268435456  Local  R/W    main
 0x600109C8  0x611BEBE1     18539034  IText  R/O    main:text
 0x611C0000  0x61642C7F      4729984  IData  R/W    main:data
 0x61642C80  0x6186607F      2241536  IBss   R/W    main:bss
 0x61866080  0x6188607F       131072  Local  R/W    main:fastheap
 0x61886080  0x6FFFFFFF    242720640  Local  R/W    main:heap
 0x80000000  0x87FFFFFF    134217728  Local  R/W    main:(main_k0)
 0x88000000  0x88001FFF         8192  Iomem  REG    qa_k0
 0x88002000  0x881FFFFF      2088960  Iomem  R/W    memd:(memd_k0)
 0xA0000000  0xA7FFFFFF    134217728  Local  R/W    main:(main_k1)
 0xA8000000  0xA8001FFF         8192  Iomem  REG    qa_k1
 0xA8002000  0xA81FFFFF      2088960  Iomem  R/W    memd:(memd_k1)
```

Related Commands

Command	Description
show context	Displays information stored in NVRAM when an unexpected system reload (system exception) occurs.
show memory	Displays detailed memory statistics for the system.
show version	Shows hardware and software information for the system.

show registry

To display the function registry information when Cisco IOS or Cisco IOS Software Modularity images are running, use the **show registry** command in user EXEC or privileged EXEC mode.

Cisco IOS Software

show registry [*registry-name* [*registry-number*]] [**brief** | **statistics**]

Cisco IOS Software Modularity

show registry [**name** [*registry-name* [*registry-number*]]] [**brief** [**name** [*registry-name* [*registry-number*]]] | **preemptions** | **rpcp status** | **statistics** [**brief**] [**name** [*registry-name* [*registry-number*]]] [**remote**]] [**process** {*process-name* | *process-id*}]

Syntax Description

Cisco IOS Software Syntax	
registry-name	(Optional) Name of the registry to display.
registry-number	(Optional) Number of the registry to display.
brief	(Optional) Displays limited functions and services information.
statistics	(Optional) Displays function registry statistics.
Cisco IOS Software Modularity Syntax	
name	(Optional) Displays information about a specific registry.
registry-name	(Optional) Name of the registry to examine.
registry-number	(Optional) Number of the registry to examine.
brief	(Optional) Displays limited functions and services information.
preemptions	(Optional) Displays registry preemptions information.
rpcp status	(Optional) Displays status of remote procedure call (RPC) proxy.
statistics	(Optional) Displays function registry statistics.
remote	(Optional) Displays name server interactions and call statistics.
process	(Optional) Displays process-specific information.
process-name	(Optional) Process name.
process-id	(Optional) Process ID. Number in range from 1 to 4294967295.

Command Default

If no options are specified, registry information is displayed for all registries.

Command Modes

User EXEC (>)
Privileged EXEC (#)

Command History

Release	Modification
11.1	This command was introduced.
12.2(18)SXF4	Keywords and arguments were added to support Software Modularity images and this command was integrated into Cisco IOS Release 12.2(18)SXF4.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Examples

Example output varies between Cisco IOS software images and Cisco IOS Software Modularity software images. To view the appropriate output, choose one of the following sections:

- Cisco IOS Software
- Cisco IOS Software Modularity

Cisco IOS Software

The following is sample output from the **show registry** command using the **brief** keyword:

```
Router# show registry atm 3/0/0 brief

Registry objects: 1799  bytes: 213412

--
Registry 23: ATM Registry
  Service 23/0:
  Service 23/1:
  Service 23/2:
  Service 23/3:
  Service 23/4:
  Service 23/5:
  Service 23/6:
  Service 23/7:
  Service 23/8:
  Service 23/9:
  Service 23/10:
  Service 23/11:
  Service 23/12:
  Service 23/13:
  Service 23/14:
.
.
.
Registry 25: ATM routing Registry
  Service 25/0:
```

Table 140 describes the significant fields shown in the display.

Table 140 show registry brief (Cisco IOS) Field Descriptions

Field	Description
Registry objects	Number of objects in the registry.
bytes	Registry size, in bytes.
Registry	Displays the specified registry service number and type of registry service.

Cisco IOS Software Modularity

The following is partial sample output from the **show registry** command when running a software Modularity image:

```
Router# show registry

Registry information for ios-base:1:
=====================================================

----------------------------
AAA_ACCOUNTING :   11 services
               /    1 :  List    list[000]
               /    2 :  List    list[000]
               /    3 :  Case    size[020] list[000] default=0x7267C5D0  returnd
               /    4 :  Case    size[020] list[000] default=0x7267C5D0  returnd
                           16  0x72779400
               /    5 :  Case    size[020] list[000] default=0x7267C5D0  returnd
               /    6 :  Case    size[020] list[000] default=0x7267C5D0  returnd
                           16  0x7277915C
               /    7 :  Retval  size[020] list[000] default=0x7267C5E4  returno
               /    8 :  Retval  size[020] list[000] default=0x7267C5E4  returno
               /    9 :  Retval  size[020] list[000] default=0x7267C5E4  returno
               /   10 :  Stub    0x7267C5E4  return_zero
               /   11 :  Stub    0x76545BA0
AAA_ACCOUNTING :   11 services,   140 global bytes,   160 heap bytes
.
.
.
```

Table 141 describes the significant fields shown in the display.

Table 141 show registry (Software Modularity) Field Descriptions

Field	Description
Registry information	Displays the registry information by process name.
services	Number of services displayed.
global bytes	Number of bytes for the service,
heap bytes	Size of the service heap, in bytes,

show reload

To display the reload status on the router, use the **show reload** command in EXEC mode.

show reload

Syntax Description

This command has no arguments or keywords.

Command Modes

EXEC

Command History

Release	Modification
11.2	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

You can use the **show reload** command to display a pending software reload. To cancel the reload, use the **reload cancel** privileged EXEC command.

Examples

The following sample output from the **show reload** command shows that a reload is schedule for 12:00 a.m. (midnight) on Saturday, April 20:

```
Router# show reload

Reload scheduled for 00:00:00 PDT Sat April 20 (in 12 hours and 12 minutes)
Router#
```

Related Commands

Command	Description
reload	Reloads the operating system.

show rom-monitor

To show both the ReadOnly and the Upgrade ROM monitor (ROMMON) image versions in addition to which ROMMON image is running on the Cisco 7200 VXR or Cisco 7301 router, use the **show rom-monitor** command in user EXEC, privileged EXEC, or diagnostic mode.

Supported Platforms Other than the Cisco ASR1000 Series Routers

show rom-monitor

Cisco ASR 1000 Series Routers

show rom-monitor *slot*

Syntax Description

slot	Specifies the slot that contains the ROMMON. Options include: • *number*—the number of the SIP slot that requires the ROMmon upgrade • **F0**—Embedded-Service-Processor slot 0 • **F1**—Embedded-Service-Processor slot 1 • **FP**—All installed Embedded-Service-Processors • **R0**—Route-Processor slot 0 • **R1**—Route-Processor slot 1 • **rp active**—active Route-Processor • **rp standby**—standby Route-Processor • **fp active**—active Embedded-Service-Processor • **fp standby**—standby Embedded-Service-Processor

Command Modes

User EXEC (>)
Privileged EXEC (#)
Diagnostic (diag)

Command History

Release	Modification
12.0(28)S	This command was introduced on the Cisco 7200 VXR router.
12.3(9)	This command was integrated into Cisco IOS Release 12.3(9) and implemented on the Cisco 7301 router.
12.3(8)T	This command was integrated into Cisco IOS Release 12.3(8)T.

Release	Modification
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.
Cisco IOS XE Release 2.1	This command was introduced on the Cisco ASR1000 Series Routers and the following enhancements were introduced: • This command was introduced in diagnostic mode for the first time. The command can be entered in both privileged EXEC and diagnostic mode on the Cisco ASR1000 Series Routers. • The *slot* keyword was introduced.

Usage Guidelines

Use the **show rom-monitor** command when you are in Cisco IOS software. Use the **showmon** command when you are in ROMMON mode.

Examples

The following sample output from the **show rom-monitor** command in Cisco IOS software, applicable to both the Cisco 7200 VXR and Cisco 7301 routers, displays both ROMMON images and verifies that the Upgrade ROMMON image is running:

```
Router> show rom-monitor

ReadOnly ROMMON version:

System Bootstrap, Version 12.2(20031011:151758)
Copyright (c) 2004 by Cisco Systems, Inc.

Upgrade ROMMON version:

System Bootstrap, Version 12.2(20031011:151758)
Copyright (c) 2004 by Cisco Systems, Inc.

Currently running ROMMON from Upgrade region
ROMMON from Upgrade region is selected for next boot
```

In the following example, the ROMmon image in RP 0 of a Cisco ASR 1006 Router is verified using the **show rom-monitor** command:

```
Router# show rom-monitor r0
System Bootstrap, Version 12.2(33r)XN1, RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 2007 by cisco Systems, Inc.
```

show rom-monitor slot

To display the ROM monitor (ROMMON) status, use the **show rom-monitor** command in user EXEC or privileged EXEC mode.

show rom-monitor slot *num* {**sp** | **rp**}

Syntax Description

num	Displays the slot number of the ROMMON for which the status is to be displayed.
sp	Displays the ROMMON status of the switch processor.
rp	Displays the ROMMON status of the route processor.

Command Modes

User EXEC
Privileged EXEC

Command History

Release	Modification
12.2(14)SX	Support for this command was introduced on the Supervisor Engine 720.
12.2(17d)SXB	Support for this command on the Supervisor Engine 2 was integrated into Release 12.2(17d)SXB.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

When you enter the **show rom-monitor slot** command, the output displays the following:

- Region region1 and region2—Displays the status of the ROMMON image and the order of preference from which the region1 or region2 images should be booted. The ROMMON image status values are as follows:
  - First run—Indicates that a check of the new image is being run.
  - Invalid—Indicates that the new image has been checked and the upgrade process has started.
  - Approved—Indicates that the ROMMON field upgrade process has completed.
- Currently running—This field displays the currently running image and the region.

The **sp** or **rp** keyword is required only if a supervisor engine is installed in the specified slot.

Examples

This example shows how to display ROMMON information:

```
Router# show rom-monitor slot 1 sp

   Region F1:APPROVED
   Region F2:FIRST_RUN, preferred
   Currently running ROMMON from F1 region
Router#
```

Related Commands

Command	Description
upgrade rom-monitor	Sets the execution preference on a ROMMON.

show running-config

To display the contents of the current running configuration file or the configuration for a specific module, Layer 2 VLAN, class map, interface, map class, policy map, or virtual circuit (VC) class, use the **show running-config** command in user EXEC or privileged EXEC mode.

show running-config [*options*]

Syntax Description

options (Optional) The following optional keywords can be entered with the **show running-config** command to customize the output according to your specific needs. Availability of these options varies by platform and Cisco IOS release. All options listed here may not be available on your specific platform and release.

- **all**—Expands the output to include the commands that are configured with default parameters. If the **all** keyword is not used, the output does not display commands configured with default parameters.
- **brief**—Displays the configuration without certification data. The **brief** keyword can be used with the **linenum** keyword.
- **class-map** *name* [**linenum**]—Displays class map information. The **linenum** keyword can be used with the **class-map** *name* option.
- **full**—Displays the full configuration. The **full** keyword can be used with the **linenum** keyword.
- **interface** *type number* [**linenum**]—Displays interface-specific configuration information. If you use the **interface** keyword, you must specify the interface type and the interface number (for example, **interface ethernet 0**). Common interfaces include **async**, **ethernet**, **fastEthernet**, **group-async**, **loopback**, **null**, **serial**, and **virtual-template**. Use the **show run interface ?** command to determine the interfaces available on your system. The **linenum** keyword can be used with the **interface** *type number* option.
- **linenum**—Displays line numbers in the output. The **brief** or **full** keyword can be used with the **linenum** keyword. The **linenum** keyword can be used with the **class-map**, **interface**, **map-class**, **policy-map**, and **vc-class** keywords.
- **map-class** [**linenum**]—Displays map class information. This option is described separately; see the **show running-config map-class** command page.
- **policy-map** *name* [**linenum**]—Displays policy map information. The **linenum** keyword can be used with the **policy-map** *name* option.
- **vc-class** *name* [**linenum**]—Displays VC class information (display is available only on certain routers such as the Cisco 7500 series—display is not available on all platforms). The **linenum** keyword can be used with the **vc-class** *name* option.
- **view full**—Enables the display of a full running configuration. This is for view-based users who typically can view only configuration commands that they are entitled to access for that particular view.
- **module** *number*—Specifies the module number.
- **vlan** *vlan-id*—Specifies the VLAN information to display; valid values are from 1 to 4094.

Command Default

The default syntax, **show running-config**, displays the contents of the running configuration file, except commands configured with default parameters.

Command Modes

User EXEC (>)
Privileged EXEC (#)

Command History

Release	Modification
11.0	This command was introduced.
12.0	This command was replaced by the **more system:running-config** command.
12.0(1)T	This command was integrated into Cisco IOS Release 12.0(1)T, and the output modifier (|) was added.
12.2(4)T	The **linenum** keyword was added.
12.3(8)T	The **view full** option was added.
12.2(14)SX	This command was integrated into Cisco IOS Release 12.2(14)SX. The **module** *number* and **vlan** *vlan-id* keywords and arguments were added for the Supervisor Engine 720.
12.2(17d)SXB	This command was integrated into Release 12.2(17d)SXB and implemented on the Supervisor Engine 2.
12.2(33)SXH	The **all** keyword was added.
12.2(31)SB2	This command was integrated into Cisco IOS Release 12.2(31)SB2. This command was enhanced to display configuration information for traffic shaping overhead accounting for ATM and was implemented on the Cisco 10000 series router for the PRE3.
12.2(33)SRC	This command was integrated into Cisco IOS Release 12.2(33)SRC.
12.2(33)SB	Support for the Cisco 7300 series router was added.

Usage Guidelines

The **show running-config** command is technically a command alias (substitute or replacement syntax) of the **more system:running-config** command. Although **more** commands are recommended (due to their uniform structure across platforms and their expandable syntax), the **show running-config** command remains enabled to accommodate its widespread use, and to allow typing shortcuts such as **show run**.

The **show running-config interface** command is useful when there are multiple interfaces and you want to look at the configuration of a specific interface.

The **linenum** keyword causes line numbers to be displayed in the output. This option is useful for identifying a particular portion of a very large configuration.

You can enter additional output modifiers in the command syntax by including a pipe character (|) after the optional keyword. For example, **show running-config interface serial 2/1 linenum | begin 3.** To display output modifiers that are available for a keyword, enter **| ?** after the keyword.

Prior to Cisco IOS Release 12.2(33)SXH, **show running-config** command output omitted configuration commands set with default values. Effective with Release 12.2(33)SXH, the **show running-config all** command displays more complete configuration information, including default settings and values. For example, if the Cisco Discovery Protocol (abbreviated as CDP in the output) holdtime value is set to its default of 180:

- The **show running-config** command does not display this value.
- The **show running-config all** displays this output: `cdp holdtime 180`.

If the Cisco Discovery Protocol holdtime is changed to a nondefault value (for example, 100), the output of the **show running-config** and **show running-config all** commands is the same; that is, the configured parameter is displayed.

Note In Release 12.2(33)SXH, implementation of the **all** keyword expands the output to include some of the commands that are configured with default values. In subsequent Cisco IOS releases, additional configuration commands that are configured with default values will be added to the output of the **show running-config all** command.

Cisco 7600 Series Router

In some cases, you might see a difference in the duplex mode that is displayed between the **show interfaces** command and the **show running-config** command. The duplex mode that is displayed in the **show interfaces** command is the actual duplex mode that the interface is running. The **show interfaces** command displays the operating mode for an interface, and the **show running-config** command displays the configured mode for an interface.

The **show running-config** command output for an interface might display the duplex mode but no configuration for the speed. This output indicates that the interface speed is configured as auto and that the duplex mode shown becomes the operational setting once the speed is configured to something other than auto. With this configuration, it is possible that the operating duplex mode for that interface does not match the duplex mode that is displayed with the **show running-config** command.

Examples

The following example shows the configuration for serial interface 1:

```
Router# show running-config interface serial 1

Building configuration...

Current configuration:
!
interface Serial1
 no ip address
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
 shutdown
end
```

The following example shows the configuration for Ethernet interface 0/0. Line numbers are displayed in the output.

```
Router# show running-config interface ethernet 0/0 linenum

Building configuration...

Current configuration : 104 bytes
 1 : !
 2 : interface Ethernet0/0
 3 :  ip address 10.4.2.63 255.255.255.0
 4 :  no ip route-cache
 5 :  no ip mroute-cache
 6 : end
```

The following example shows how to set line numbers in the command output and then use the output modifier to start the display at line 10:

```
Router# show running-config linenum | begin 10

   10 : boot-start-marker
   11 : boot-end-marker
   12 : !
   13 : no logging buffered
   14 : enable password #####
   15 : !
   16 : spe 1/0 1/7
   17 :  firmware location bootflash:mica-modem-pw.172.16.0.0.bin
   18 : !
   19 : !
   20 : resource-pool disable
   21 : !
   22 : no aaa new-model
   23 : ip subnet-zero
   24 : ip domain name cisco.com
   25 : ip name-server 172.16.11.48
   26 : ip name-server 172.16.2.133
   27 : !
   28 : !
   29 : isdn switch-type primary-5ess
   30 : !
   .
   .
   .
   126 : end
```

The following example shows how to display the module and status configuration for all modules on a Cisco 7600 series router:

```
Router# show running-config

Building configuration...

Current configuration:
!
version 12.0
service timestamps debug datetime localtime
service timestamps log datetime localtime
no service password-encryption
!
hostname Router
!
boot buffersize 126968
boot system flash slot0:7600r
boot bootldr bootflash:c6msfc-boot-mz.120-6.5T.XE1.0.83.bin
enable password lab
!
clock timezone Pacific -8
clock summer-time Daylight recurring
redundancy
 main-cpu
  auto-sync standard
!
ip subnet-zero
!
ip multicast-routing
ip dvmrp route-limit 20000
ip cef
```

```
mls flow ip destination
mls flow ipx destination
cns event-service server
!
spanning-tree portfast bpdu-guard
spanning-tree uplinkfast
spanning-tree vlan 200 forward-time 21
port-channel load-balance sdip
!
!
!
 shutdown
!
!
.
.
.
```

In the following sample output from the **show running-config** command, the **shape average** command indicates that traffic shaping overhead accounting for ATM is enabled. The BRAS-DSLAM encapsulation type is qinq and the subscriber line encapsulation type is snap-rbe based on the AAL5 service.

```
Router# show running-config
.
.
.
subscriber policy recording rules limit 64
no mpls traffic-eng auto-bw timers frequency 0
call rsvp-sync
!
controller T1 2/0
    framing sf
    linecode ami
!
controller T1 2/1
    framing sf
    linecode ami
!
!
policy-map unit-test
    class class-default
        shape average percent 10 account qinq aal5 snap-rbe
!
```

Related Commands

Command	Description
bandwidth	Specifies or modifies the bandwidth allocated for a class belonging to a policy map, and enables ATM overhead accounting.
boot config	Specifies the device and filename of the configuration file from which the router configures itself during initialization (startup).
configure terminal	Enters global configuration mode.
copy running-config startup-config	Copies the running configuration to the startup configuration. (Command alias for the **copy system:running-config nvram:startup-config** command.)
shape	Shapes traffic to the indicated bit rate according to the algorithm specified, and enables ATM overhead accounting.
show interfaces	Displays statistics for all interfaces configured on the router or access server.

Command	Description
show policy-map	Displays the configuration of all classes for a specified service policy map or all classes for all existing policy maps, and displays ATM overhead accounting information, if configured.
show startup-config	Displays the contents of NVRAM (if present and valid) or displays the configuration file pointed to by the CONFIG_FILE environment variable. (Command alias for the **more:nvram startup-config** command.)

show running-config map-class

To display only map-class configuration information from the running configuration file, use the **show running-config map-class** command in privileged EXEC mode.

show running-config map-class [**atm** [*map-class-name*] | **dialer** [*map-class-name*] | **frame-relay** [*map-class-name*]] [**linenum**]

Syntax Description

atm	(Optional) Displays only ATM map-class configuration lines.
dialer	(Optional) Displays only dialer map-class configuration lines.
frame-relay	(Optional) Displays only Frame Relay map-class configuration lines.
map-class-name	(Optional) Displays only configuration lines for the specified map-class.
linenum	(Optional) Displays line numbers in the output.

Defaults

Displays all map-class configuration in the running configuration file.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1	The **map-class** extension to the **show running-config** command was introduced to show only lines pertaining to dialer or Frame Relay map classes.
12.1(2)T	The **atm**, **dialer**, and **frame-relay** keywords and *map-class-name* argument were introduced.
12.2(4)T	The **linenum** keyword was added.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

Use the **show running-config map-class** command to display the following information from the running configuration file:

- All map classes configured on the router.
- Map classes configured specifically for ATM, Frame Relay, or dialer.
- A specific ATM, Frame Relay, or dialer map class.

Use the **linenum** keyword to display line numbers in the output. This option is useful for identifying a particular portion of a very large configuration.

Examples

All Map Classes Configured on the Router Example

The following example displays all map classes configured on the router:

```
Router# show running-config map-class
```

```
Building configuration...
Current configuration:
!
map-class frame-relay cir60
 frame-relay bc 16000
 frame-relay adaptive-shaping becn
!
map-class frame-relay cir70
 no frame-relay adaptive-shaping
 frame-relay priority-group 2
!
map-class atm vc100
 atm aal5mux
!
map-class dialer dialer1
 dialer idle-timeout 10
end
```

All Frame Relay Map Classes Example

The following example displays all Frame Relay map classes on the router:

```
Router# show running-config map-class frame-relay

Building configuration...
Current configuration:
!
map-class frame-relay cir60
 frame-relay bc 16000
 frame-relay adaptive-shaping becn
!
map-class frame-relay cir70
 no frame-relay adaptive-shaping
 frame-relay priority-group 2
end
```

A Specific Map Class and Display of Line Numbers Example

The following example displays a specific map class called class1. Line numbers are displayed in the output.

```
Router# show running-config map-class frame-relay class1 linenum

Building configuration...

Current configuration:
1 : !
2 : map-class frame-relay boy
3 :  no frame-relay adaptive-shaping
4 :  frame-relay cir 1000
5 : end
```

Related Commands

Command	Description
map-class atm	Specifies the ATM map class for an SVC.
map-class dialer	Defines a class of shared configuration parameters associated with the dialer map command for outgoing calls from an ISDN interface and for PPP callback.

Command	Description
map-class frame-relay	Specifies a map class to define QoS values for a Frame Relay VC.
more system:running-config	Displays contents of the currently running configuration file (equivalent to the **show running-config** command.)

show running-config partition

To display the list of commands that make up the current running configuration for a specific part of the system's global running configuration, use the **show running-config partition** command in privileged EXEC mode.

show running-config partition *part*

Syntax Description

part	The *part* argument will consist of one or more keyword options. These keywords represent a partition of the system's running configuration state, as a major-descriptor and, in some cases, one or more minor-descriptors. For example, in the command **show running-config partition router eigrp 1**, the major-descriptor for the *part* argument is the **router** keyword, and the minor-descriptors for the *part* argument are the **eigrp 1** keywords. The actual list of *part* keyword options will depend on your system hardware, what feature set you are running, and what features are currently configured on your system. Some examples of command *part* keyword options are provided here for reference. Use the **show running-config partition ?** command on your system to view the list of command options available on your system. • **access-list**—Displays all running configuration commands that make up the access-list configuration partition. • **boot**—Displays all running configuration commands that make up the boot configuration partition. • **class-map**—Displays all running configuration commands that make up the class-map configuration partition. • **global-cdp**—Displays all running configuration commands that make up the global CDP configuration partition. • **interface** [**type** *slot*/*port*/*number*]—Displays all running configuration commands that make up the interfaces configuration partition or the configuration commands that are applied to the specified interface. • **line**—Displays all running configuration commands that make up the line command configuration partition. • **policy-map**—Displays all running configuration commands that make up the policy-map configuration partition. • **route-map**—Displays all running configuration commands that make up the route-map configuration partition. • **router** [*protocol*]—Displays all running configuration commands that make up the router configuration partition, or the configuration commands for the specified routing protocol. • **service**—Displays all running configuration commands that make up the services (small server) configuration partition. • **snmp**—Displays all running configuration commands that make up the SNMP configuration partition. • **| –** Allows for the addition of output modifiers.

Command Default None

Command Modes Privileged EXEC (#)

Command History

Release	Modification
12.2(33)SRB	This command was introduced for Cisco 7600 series images in Cisco IOS Release 12.2SR as part of the "Configuration Partitioning" feature.
12.2(33)SB	This command was integrated into Cisco IOS Release 12.2(33)SB and implemented on the Cisco 10000 series.

Usage Guidelines When the Configuration Partitioning feature is enabled, the system groups the configuration state of the device into parts (called "partitions") for the purpose of generating the virtual running configuration file (the list of configuration commands). The selective processing of the system's configuration state for the purpose of generating a partial running configuration is called "configuration partitioning."

Note This command is not related to hard drive or flash drive partitioning.

This granular access to configuration information offers important performance benefits for high-end routing platforms with very large configuration files, as the system wide generation of a complete virtual configuration file from all components on systems with large and complex configurations can become overly resource intensive and be unacceptably slow.

The **show running-config partition** command allows you to display only the part of the running configuration that you want to examine, while also allowing the system to process only the collection of system components (such as specific interfaces) that you need to display. This is in contrast to other existing extensions to the **show running-config** command, which only *filter* the generated list after all system components have been processed.

The Configuration Partitioning feature is enabled by default in Cisco IOS software images that support the feature. To disable the feature, use the **no parser config partition** command.

Examples In the following example, the system generates a view of the running configuration by polling only the components associated with the access-list parts of the running configuration state, and then displays only those access-list-related configuration commands.

```
Router# show running-config partition access-list
        Building configuration...

Current configuration : 127 bytes
!
Configuration of Partition access-list
!
!
!
access-list 90 permit 0.0.0.0 1.2.3.5
access-list 100 permit 10 any any
!
end
```

In the following example, only the main configuration partition associated with the interface configuration is queried, and only the configuration commands associated with FastEthernet interface 0/1 are displayed.

```
Router# show running-config partition interface fastethernet0/1
Building configuration...

Current configuration : 213 bytes
!
Configuration of Partition interface FastEthernet0/1
!
!
interface FastEthernet0/1
 ip address 10.4.2.39 255.255.255.0
 no ip route-cache cef
 no ip route-cache
 duplex half
 ipv6 enable
 no cdp enable
!
!
end
```

Related Commands

Command	Description
copy running-config startup-config	Copies the running configuration to the default startup configuration file.
show interfaces	Displays statistics for all interfaces configured on the router or access server.
show running-config	Generates and displays a virtual configuration file that lists all configuration commands that are in effect on the system.
show startup-config	Displays the contents of NVRAM (if present and valid) or displays the configuration file pointed to by the CONFIG_FILE environment variable. (Command alias for the **more:nvram startup-config** command.)

show scp

To display Switch-Module Configuration Protocol (SCP) information, use the **show scp** in privileged EXEC mode on the Switch Processor.

show scp {**accounting** | **counters** | **linecards** [**details**] | **mcast** {**group** *group-id* | **inst**} | **process** *id* | **status**}

Syntax Description

accounting	Displays information about the SCP accounting.
counters	Displays information about the SCP counter.
linecards	Displays information about the Optical Services Module (OSM) wide area network (WAN) modules in the chassis.
details	(Optional) Displays detailed information about the OSM WAN module.
mcast	Displays information about the SCP multicast.
group *group-id*	(Optional) Displays information for a specific group and group ID; valid values are from 1 to 127.
inst	(Optional) Displays information for an instance.
process *id*	Displays all the processes that have registered an SAP with SCP.
status	Displays information about the local SCP server status.

Defaults

This command has no default settings.

Command Modes

Privileged EXEC on the Switch Processor

Command History

Release	Modification
12.2(14)SX	Support for this command was introduced on the Supervisor Engine 720.
12.2(17d)SXB	Support for this command on the Supervisor Engine 2 was extended to Release 12.2(17d)SXB.
12.2(18)SXE	The output of the **show scp process** command was changed to display all the processes that have registered an SAP with SCP on the Supervisor Engine 720 only.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Examples

This example shows how to display all the processes that have registered an SAP with SCP:

```
Router# show module

Mod Ports Card Type                                Model              Serial No.
--- ----- ---------------------------------------- ------------------ -----------
  1   48  48-port 10/100 mb RJ45                   WS-X6148-RJ-45     SAL091800RY
  2    0  2 port adapter Enhanced FlexWAN          WS-X6582-2PA       JAE0940MH7Z
  3    8  8 port 1000mb GBIC Enhanced QoS          WS-X6408A-GBIC     SAL09391KZH
  5    2  Supervisor Engine 720 (Active)           WS-SUP720-3BXL     SAL09337UE6
```

```
  6    2  Supervisor Engine 720 (Hot)            WS-SUP720-3BXL     SAL09148P59

Mod MAC addresses                       Hw    Fw           Sw           Status
--- ----------------------------------- ------ ------------ ------------ -------
  1  0013.c3f8.d2c4 to 0013.c3f8.d2f3   5.0   8.3(1)       8.6(0.366)TA Ok
  2  0015.2bc3.5b40 to 0015.2bc3.5b7f   2.1   12.2(nightly 12.2(nightly Ok
  3  0015.6324.ed48 to 0015.6324.ed4f   3.1   5.4(2)       8.6(0.366)TA Ok
  5  0014.a97d.b0ac to 0014.a97d.b0af   4.3   8.4(2)       12.2(nightly Ok
  6  0013.7f0d.0660 to 0013.7f0d.0663   4.3   8.4(2)       12.2(nightly Ok

Mod  Sub-Module                  Model              Serial       Hw     Status
---- --------------------------- ------------------ ------------ ------- -------
  5  Policy Feature Card 3       WS-F6K-PFC3BXL     SAL09337NVE  1.6    Ok
  5  MSFC3 Daughterboard         WS-SUP720          SAL09327AU6  2.3    Ok
  6  Policy Feature Card 3       WS-F6K-PFC3BXL     SAL1033Y0YK  1.8    Ok
  6  MSFC3 Daughterboard         WS-SUP720          SAL09158XB3  2.3    Ok

Mod  Online Diag Status
---- -------------------
  1  Pass
  2  Pass
  3  Pass
  5  Pass
  6  Pass


Router# attach 5

Trying Switch ...
Entering CONSOLE for Switch
Type "^C^C^C" to end this session

Switch-sp# show scp process

Sap Pid Name
=== === ====
0 180 CWAN-RP SCP Input Process
18 42 itasca
20 3 Exec
21 3 Exec
22 180 CWAN-RP SCP Input Process
Total number of SAP registered = 5
Router#
```

show slot

To display information about the PCMCIA flash memory cards file system, use the **show slot** command in user EXEC or privileged EXEC mode.

show slot [**all** | **chips** | **detailed** | **err** | **summary**]

Syntax Description

all	(Optional) Displays all possible flash system information for all PCMCIA flash cards in the system.
chips	(Optional) Displays flash chip information.
detailed	(Optional) Displays the flash detailed directory.
err	(Optional) Displays the flash chip erase and write retries.
summary	(Optional) Displays the flash partition summary.

Command Modes

User EXEC
Privileged EXEC

Command History

Release	Modification
12.0	This command was introduced.

Usage Guidelines

Use the **show slot** command to display details about the files in a particular linear PCMCIA flash memory card of less than 20 MB and some 32 MB linear PCMCIA cards.

Note Use the **show disk** command for ATA PCMCIA cards. Other forms of this commands are **show disk0:** and **show disk1:**.

For more information regarding file systems and flash cards, access the *PCMCIA Filesystem Compatibility Matrix and Filesystem Information* document at the following URL:

http://www.cisco.com/en/US/partner/products/hw/routers/ps341/products_tech_note09186a00800a7515.shtml

To see which flash cards are used in your router, use the **show version** command and look at the bottom portion of the output.

The following display indicates an ATA PCMCIA flash disk.

```
Router# show version
.
.
46976K bytes of ATA PCMCIA card at slot 0 (Sector size 512 bytes).
```

The following display indicates a linear PCMCIA flash card with 20480K bytes of flash memory in card at slot 1 with a sector size of 128K.

```
Router# show version
.
.
20480K bytes of Flash PCMCIA card at slot 1 (Sector size 128K).
```

Note In some cases the **show slot** command will not display the file systems, use **show slot0:** or **show slot1:**.

Examples

The following example displays information about slot 0. The output is self-explanatory.

```
Router# show slot

PCMCIA Slot0 flash directory:
File  Length   Name/status
  1   11081464  c3660-bin-mz.123-9.3.PI5b
[11081528 bytes used, 9627844 available, 20709372 total]
20480K bytes of processor board PCMCIA Slot0 flash (Read/Write)
```

The following example shows all possible flash system information for all PCMCIA flash cards in the system.

```
Router# show slot all
Partition   Size    Used      Free      Bank-Size  State          Copy Mode
  1        20223K  10821K     9402K     4096K      Read/Write     Direct

PCMCIA Slot0 flash directory:
File  Length   Name/status
        addr      fcksum  ccksum
  1   11081464  c3660-bin-mz.123-9.3.PI5b
        0x40      0x5EA3  0x5EA3
[11081528 bytes used, 9627844 available, 20709372 total]
20480K bytes of processor board PCMCIA Slot0 flash (Read/Write)

   Chip    Bank    Code      Size      Name
    1      1       89A0      2048KB    INTEL 28F016SA
    2      1       89A0      2048KB    INTEL 28F016SA
    1      2       89A0      2048KB    INTEL 28F016SA
    2      2       89A0      2048KB    INTEL 28F016SA
    1      3       89A0      2048KB    INTEL 28F016SA
    2      3       89A0      2048KB    INTEL 28F016SA
    1      4       89A0      2048KB    INTEL 28F016SA
    2      4       89A0      2048KB    INTEL 28F016SA
    1      5       89A0      2048KB    INTEL 28F016SA
    2      5       89A0      2048KB    INTEL 28F016SA


The following example shows flash chip information

Router# show slot chips
20480K bytes of processor board PCMCIA Slot0 flash (Read/Write)

   Chip    Bank    Code      Size      Name
    1      1       89A0      2048KB    INTEL 28F016SA
    2      1       89A0      2048KB    INTEL 28F016SA
    1      2       89A0      2048KB    INTEL 28F016SA
    2      2       89A0      2048KB    INTEL 28F016SA
    1      3       89A0      2048KB    INTEL 28F016SA
    2      3       89A0      2048KB    INTEL 28F016SA
    1      4       89A0      2048KB    INTEL 28F016SA
    2      4       89A0      2048KB    INTEL 28F016SA
    1      5       89A0      2048KB    INTEL 28F016SA
    2      5       89A0      2048KB    INTEL 28F016SA
```

The following example show the flash detailed directory.

```
Router# show slot detailed

PCMCIA Slot0 flash directory:
File  Length   Name/status
        addr      fcksum  ccksum
  1   11081464  c3660-bin-mz.123-9.3.PI5b
        0x40      0x5EA3  0x5EA3
[11081528 bytes used, 9627844 available, 20709372 total]
20480K bytes of processor board PCMCIA Slot0 flash (Read/Write)
```

The following example shows the flash chip erase and write retries.

```
Router# show slot err

PCMCIA Slot0 flash directory:
File  Length   Name/status
  1   11081464  c3660-bin-mz.123-9.3.PI5b
[11081528 bytes used, 9627844 available, 20709372 total]
20480K bytes of processor board PCMCIA Slot0 flash (Read/Write)

   Chip    Bank    Code      Size      Name               erase  write
    1      1       89A0      2048KB    INTEL 28F016SA     0      0
    2      1       89A0      2048KB    INTEL 28F016SA     0      0
    1      2       89A0      2048KB    INTEL 28F016SA     0      0
    2      2       89A0      2048KB    INTEL 28F016SA     0      0
    1      3       89A0      2048KB    INTEL 28F016SA     0      0
    2      3       89A0      2048KB    INTEL 28F016SA     0      0
    1      4       89A0      2048KB    INTEL 28F016SA     0      0
    2      4       89A0      2048KB    INTEL 28F016SA     0      0
    1      5       89A0      2048KB    INTEL 28F016SA     0      0
    2      5       89A0      2048KB    INTEL 28F016SA     0      0
```

The following example shows the flash partition summary.

```
Router# show slot summary
Partition   Size    Used      Free      Bank-Size  State          Copy Mode
  1        20223K  10821K     9402K     4096K      Read/Write     Direct
20480K bytes of processor board PCMCIA Slot0 flash (Read/Write)
```

Related Commands

Command	Description
dir slot0:	Directory listing of files on a PCMCIA Flash card located in slot0.
dir slot1:	Directory listing of files on a PCMCIA Flash card located in slot1.
show slot0:	Displays information about the PCMCIA flash memory card's file system located in slot 0.
show slot1:	Displays information about the PCMCIA flash memory card's file system located in slot 1.

show slot0:

To display information about the PCMCIA flash memory card's file system located in slot 0, use the **show slot0:** command in user EXEC or privileged EXEC mode.

show slot0: [**all** | **chips** | **detailed** | **err** | **summary**]

Syntax Description

all	(Optional) Displays all possible flash system information for all PCMCIA flash cards in the system.
chips	(Optional) Displays flash chip information.
detailed	(Optional) Displays the flash detailed directory.
err	(Optional) Displays the flash chip erase and write retries.
summary	(Optional) Displays the flash partition summary.

Command Modes

User EXEC
Privileged EXEC

Command History

Release	Modification
12.0	This command was introduced.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

Use the **show slot0:** command to display details about the files in a particular linear PCMCIA flash memory card of less than 20 MB and some 32 MB linear PCMCIA cards.

Note Use the **show disk** command for ATA PCMCIA cards. Other forms of this commands are **show disk0:** and **show disk1:**.

For more information regarding file systems and flash cards, access the *PCMCIA Filesystem Compatibility Matrix and Filesystem Information* document at the following URL:

http://www.cisco.com/en/US/partner/products/hw/routers/ps341/products_tech_note09186a00800a7515.shtml

To see which flash cards are used in your router, use the **show version** command and look at the bottom portion of the output.

The following display indicates an ATA PCMCIA flash disk.

```
Router# show version
.
.
46976K bytes of ATA PCMCIA card at slot 0 (Sector size 512 bytes).
```

The following display indicates a linear PCMCIA flash card with 20480K bytes of flash memory in card at slot 1 with a sector size of 128K.

```
Router# show version
.
.
20480K bytes of Flash PCMCIA card at slot 1 (Sector size 128K).
```

Note In some cases the **show slot** command will not display the file systems, use **show slot0:** or **show slot1:**.

Examples

The following example displays information about slot 0. The output is self-explanatory.

```
Router# show slot0:

PCMCIA Slot0 flash directory:
File  Length   Name/status
  1   11081464  c3660-bin-mz.123-9.3.PI5b
[11081528 bytes used, 9627844 available, 20709372 total]
20480K bytes of processor board PCMCIA Slot0 flash (Read/Write)

Router# show slot0: all
Partition   Size    Used      Free      Bank-Size  State          Copy Mode
  1        20223K  10821K     9402K     4096K      Read/Write     Direct

PCMCIA Slot0 flash directory:
File  Length   Name/status
        addr      fcksum  ccksum
  1   11081464  c3660-bin-mz.123-9.3.PI5b
        0x40      0x5EA3  0x5EA3
[11081528 bytes used, 9627844 available, 20709372 total]
20480K bytes of processor board PCMCIA Slot0 flash (Read/Write)

   Chip    Bank    Code      Size      Name
    1      1       89A0      2048KB    INTEL 28F016SA
    2      1       89A0      2048KB    INTEL 28F016SA
    1      2       89A0      2048KB    INTEL 28F016SA
    2      2       89A0      2048KB    INTEL 28F016SA
    1      3       89A0      2048KB    INTEL 28F016SA
    2      3       89A0      2048KB    INTEL 28F016SA
    1      4       89A0      2048KB    INTEL 28F016SA
    2      4       89A0      2048KB    INTEL 28F016SA
    1      5       89A0      2048KB    INTEL 28F016SA
    2      5       89A0      2048KB    INTEL 28F016SA

The following example shows flash chip information.

Router# show slot0: chips
20480K bytes of processor board PCMCIA Slot0 flash (Read/Write)

   Chip    Bank    Code      Size      Name
    1      1       89A0      2048KB    INTEL 28F016SA
    2      1       89A0      2048KB    INTEL 28F016SA
    1      2       89A0      2048KB    INTEL 28F016SA
    2      2       89A0      2048KB    INTEL 28F016SA
    1      3       89A0      2048KB    INTEL 28F016SA
    2      3       89A0      2048KB    INTEL 28F016SA
    1      4       89A0      2048KB    INTEL 28F016SA
    2      4       89A0      2048KB    INTEL 28F016SA
    1      5       89A0      2048KB    INTEL 28F016SA
    2      5       89A0      2048KB    INTEL 28F016SA
```

The following example show the flash detailed directory.

```
Router# show slot0: detailed
```

```
PCMCIA Slot0 flash directory:
File  Length   Name/status
        addr      fcksum  ccksum
  1   11081464  c3660-bin-mz.123-9.3.PI5b
        0x40      0x5EA3  0x5EA3
[11081528 bytes used, 9627844 available, 20709372 total]
20480K bytes of processor board PCMCIA Slot0 flash (Read/Write)
```

The following example shows the flash chip erase and write retries.

```
Router# show slot0: err

PCMCIA Slot0 flash directory:
File  Length   Name/status
  1   11081464  c3660-bin-mz.123-9.3.PI5b
[11081528 bytes used, 9627844 available, 20709372 total]
20480K bytes of processor board PCMCIA Slot0 flash (Read/Write)

   Chip    Bank    Code      Size      Name               erase  write
    1      1       89A0      2048KB    INTEL 28F016SA     0      0
    2      1       89A0      2048KB    INTEL 28F016SA     0      0
    1      2       89A0      2048KB    INTEL 28F016SA     0      0
    2      2       89A0      2048KB    INTEL 28F016SA     0      0
    1      3       89A0      2048KB    INTEL 28F016SA     0      0
    2      3       89A0      2048KB    INTEL 28F016SA     0      0
    1      4       89A0      2048KB    INTEL 28F016SA     0      0
    2      4       89A0      2048KB    INTEL 28F016SA     0      0
    1      5       89A0      2048KB    INTEL 28F016SA     0      0
    2      5       89A0      2048KB    INTEL 28F016SA     0      0
```

The following example shows the flash partition summary.

```
Router# show slot0: summary
Partition   Size    Used      Free      Bank-Size  State          Copy Mode
  1         20223K  10821K    9402K     4096K      Read/Write     Direct
20480K bytes of processor board PCMCIA Slot0 flash (Read/Write)
```

Related Commands

Command	Description
dir slot0:	Directory listing of files on a PCMCIA Flash card located in slot0.
dir slot1:	Directory listing of files on a PCMCIA Flash card located in slot1.
show slot1:	Displays information about the PCMCIA flash memory card's file system located in slot 1.
show slot	Displays information about the PCMCIA flash memory cards.

show slot1:

To display information about the PCMCIA flash memory card's file system located in slot 1, use the **show slot1:** command in user EXEC or privileged EXEC mode.

show slot1: [**all** | **chips** | **detailed** | **err** | **summary**]

Syntax Description

all	(Optional) Displays all possible flash system information for all PCMCIA flash cards in the system.
chips	(Optional) Displays flash chip information.
detailed	(Optional) Displays the flash detailed directory.
err	(Optional) Displays the flash chip erase and write retries.
summary	(Optional) Displays the flash partition summary.

Command Modes

User EXEC
Privileged EXEC

Command History

Release	Modification
12.0	This command was introduced.

Usage Guidelines

Use the **show slot1:** command to display details about the files in a particular linear PCMCIA flash memory card of less than 20 MB and some 32 MB linear PCMCIA cards located in slot 1.

Note Use the **show disk** command for ATA PCMCIA cards. Other forms of this commands are **show disk0:** and **show disk1:**.

For more information regarding file systems and flash cards, access the *PCMCIA Filesystem Compatibility Matrix and Filesystem Information* document at the following URL:

http://www.cisco.com/en/US/partner/products/hw/routers/ps341/products_tech_note09186a00800a7515.shtml

To see which flash cards are used in your router, use the **show version** command and look at the bottom portion of the output.

The following display indicates an ATA PCMCIA flash disk.

```
Router# show version
.
.
46976K bytes of ATA PCMCIA card at slot 0 (Sector size 512 bytes).
```

The following display indicates a linear PCMCIA flash card with 20480K bytes of flash memory in card at slot 1 with a sector size of 128K.

```
Router# show version
.
.
20480K bytes of Flash PCMCIA card at slot 1 (Sector size 128K).
```

Note In some cases the **show slot** command will not display the file systems. Use **show slot0:** or **show slot1:**.

Examples

The following example displays information about slot 0 using the **slot0:** command form. The output is self-explanatory.

```
Router# show slot1:

PCMCIA Slot1 flash directory:
File  Length   Name/status
  1   10907068  c3660-bin-mz.123-7.9.PI4
[10907132 bytes used, 5739008 available, 16646140 total]
16384K bytes of processor board PCMCIA Slot1 flash (Read/Write)

Router# show slot1: all
Partition   Size    Used      Free      Bank-Size  State          Copy Mode
  1        20223K  10821K     9402K     4096K      Read/Write     Direct

PCMCIA Slot0 flash directory:
File  Length   Name/status
        addr      fcksum  ccksum
  1   11081464  c3660-bin-mz.123-9.3.PI5b
        0x40      0x5EA3  0x5EA3
[11081528 bytes used, 9627844 available, 20709372 total]
20480K bytes of processor board PCMCIA Slot0 flash (Read/Write)

   Chip    Bank    Code      Size      Name
    1      1       89A0      2048KB    INTEL 28F016SA
    2      1       89A0      2048KB    INTEL 28F016SA
    1      2       89A0      2048KB    INTEL 28F016SA
    2      2       89A0      2048KB    INTEL 28F016SA
    1      3       89A0      2048KB    INTEL 28F016SA
    2      3       89A0      2048KB    INTEL 28F016SA
    1      4       89A0      2048KB    INTEL 28F016SA
    2      4       89A0      2048KB    INTEL 28F016SA
    1      5       89A0      2048KB    INTEL 28F016SA
    2      5       89A0      2048KB    INTEL 28F016SA
```

The following example shows flash chip information.

```
Router# show slot1: chips
20480K bytes of processor board PCMCIA Slot0 flash (Read/Write)

   Chip    Bank    Code      Size      Name
    1      1       89A0      2048KB    INTEL 28F016SA
    2      1       89A0      2048KB    INTEL 28F016SA
    1      2       89A0      2048KB    INTEL 28F016SA
    2      2       89A0      2048KB    INTEL 28F016SA
    1      3       89A0      2048KB    INTEL 28F016SA
    2      3       89A0      2048KB    INTEL 28F016SA
    1      4       89A0      2048KB    INTEL 28F016SA
    2      4       89A0      2048KB    INTEL 28F016SA
    1      5       89A0      2048KB    INTEL 28F016SA
    2      5       89A0      2048KB    INTEL 28F016SA
```

The following example show the flash detailed directory.

```
Router# show slot1: detailed

PCMCIA Slot0 flash directory:
```

```
File  Length   Name/status
        addr      fcksum  ccksum
  1   11081464  c3660-bin-mz.123-9.3.PI5b
        0x40      0x5EA3  0x5EA3
[11081528 bytes used, 9627844 available, 20709372 total]
20480K bytes of processor board PCMCIA Slot0 flash (Read/Write)
```

The following example shows the flash chip erase and write retries.

```
Router# show slot1: err

PCMCIA Slot0 flash directory:
File  Length   Name/status
  1   11081464  c3660-bin-mz.123-9.3.PI5b
[11081528 bytes used, 9627844 available, 20709372 total]
20480K bytes of processor board PCMCIA Slot0 flash (Read/Write)

   Chip    Bank    Code      Size      Name                erase  write
    1      1       89A0      2048KB    INTEL 28F016SA      0      0
    2      1       89A0      2048KB    INTEL 28F016SA      0      0
    1      2       89A0      2048KB    INTEL 28F016SA      0      0
    2      2       89A0      2048KB    INTEL 28F016SA      0      0
    1      3       89A0      2048KB    INTEL 28F016SA      0      0
    2      3       89A0      2048KB    INTEL 28F016SA      0      0
    1      4       89A0      2048KB    INTEL 28F016SA      0      0
    2      4       89A0      2048KB    INTEL 28F016SA      0      0
    1      5       89A0      2048KB    INTEL 28F016SA      0      0
    2      5       89A0      2048KB    INTEL 28F016SA      0      0
```

The following example shows the flash partition summary.

```
Router# show slot1: summary
Partition   Size    Used      Free      Bank-Size  State          Copy Mode
  1        20223K  10821K     9402K     4096K      Read/Write     Direct
20480K bytes of processor board PCMCIA Slot0 flash (Read/Write)
```

Related Commands

Command	Description
dir slot0:	Directory listing of files on a PCMCIA Flash card located in slot0.
dir slot1:	Directory listing of files on a PCMCIA Flash card located in slot1.
show slot0:	Displays information about the PCMCIA flash memory card's file system located in slot 0.
show slot	Displays information about the PCMCIA flash memory cards.

show stacks

To monitor the stack usage of processes and interrupt routines, use the **show stacks** command in EXEC mode.

show stacks

Syntax Description

This command has no arguments or keywords.

Command Modes

EXEC

Command History

Release	Modification
10.0	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

The display from this command includes the reason for the last system reboot. If the system was reloaded because of a system failure, a saved system stack trace is displayed. This information is of use only to your technical support representative in analyzing crashes in the field. It is included here in case you need to read the displayed statistics to an engineer over the phone.

Examples

The following is sample output from the **show stacks** command following a system failure:

```
Router# show stacks

Minimum process stacks:
Free/Size  Name
 652/1000  Router Init
 726/1000  Init
 744/1000  BGP Open
 686/1200  Virtual Exec

Interrupt level stacks:
Level    Called Free/Size  Name
  1           0 1000/1000  env-flash
  3         738  900/1000  Multiport Communications Interfaces
  5         178  970/1000  Console UART
System was restarted by bus error at PC 0xAD1F4, address 0xD0D0D1A
GS Software (GS3), Version 9.1(0.16), BETA TEST SOFTWARE
Compiled Tue 11-Aug-92 13:27 by jthomas
Stack trace from system failure:
FP: 0x29C158, RA: 0xACFD4
FP: 0x29C184, RA: 0xAD20C
FP: 0x29C1B0, RA: 0xACFD4
FP: 0x29C1DC, RA: 0xAD304
FP: 0x29C1F8, RA: 0xAF774
FP: 0x29C214, RA: 0xAF83E
FP: 0x29C228, RA: 0x3E0CA
FP: 0x29C244, RA: 0x3BD3C
```

Related Commands

Command	Description
show processes	Displays information about the active processes.

show startup-config

The **more nvram:startup-config** command has been replaced by the **show startup-config** command. See the description of the **more** command in the "Cisco IOS File System Commands" chapter for more information.

show subsys

To display the subsystem information, use the **show subsys** command in privileged EXEC mode.

show subsys [**class** *class* | **name** *name*]

Syntax Description

class *class*	(Optional) Displays the subsystems of the specified class. Valid classes are **driver**, **kernel**, **library**, **license**, **management**, **protocol**, and **registry**.
name *name*	(Optional) Displays the specified subsystem. Use the asterisk character (*) as a wildcard at the end of the name to list all subsystems, starting with the specified characters.

Command Modes

Privileged EXEC

Command History

Release	Modification
11.1	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(35)SE2	The **license** class was added, and this command was integrated into Cisco IOS Release 12.2(35)SE1.

Usage Guidelines

Use the **show subsys** command to confirm that all required features are in the running image.

Examples

Following is sample output from the **show subsys** command:

```
Router# show subsys

Name                Class       Version
static_map          Kernel      1.000.001
arp                 Kernel      1.000.001
ether               Kernel      1.000.001
compress            Kernel      1.000.001
alignment           Kernel      1.000.002
monvar              Kernel      1.000.001
slot                Kernel      1.000.001
oir                 Kernel      1.000.001
atm                 Kernel      1.000.001
ip_addrpool_sys     Library     1.000.001
chat                Library     1.000.001
dialer              Library     1.000.001
flash_services      Library     1.000.001
ip_localpool_sys    Library     1.000.001
nvram_common        Driver      1.000.001
ASP                 Driver      1.000.001
sonict              Driver      1.000.001
oc3suni             Driver      1.000.001
oc12suni            Driver      1.000.001
ds3suni             Driver      1.000.001
```

Following is sample output from the **show subsys** command that includes the **license** class:

```
Router# show subsys

Name                Class       Version
license_mgmt_local  Management  1.000.001
license_admin_local Management  1.000.001
license_debug_core  Management  1.000.001
license_test_ui     Management  1.000.001
test_license_parser Management  1.000.001
license_ui          Management  1.000.001
license_parser      Management  1.000.001
license_registry    Registry    1.000.001
license_client      License     1.000.001
```

Table 142 describes the fields shown in the display.

Table 142 show subsys Field Descriptions

Field	Description
Name	Name of the subsystem.
Class	Class of the subsystem. Possible classes include Driver, Kernel, Library, License, Management, Protocol, Registry.
Version	Version of the subsystem.

show sup-bootflash

To display information about the sup-bootflash file system, use the **show sup-bootflash** command in privileged EXEC mode.

show sup-bootflash [**all** | **chips** | **filesys**]

Syntax Description

all	(Optional) Displays all possible Flash information.
chips	(Optional) Displays information about the Flash chip.
filesys	(Optional) Displays information about the file system.

Defaults

This command has no default settings.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(14)SX	Support for this command was introduced on the Supervisor Engine 720.
12.2(17d)SXB	Support for this command on the Supervisor Engine 2 was extended to Release 12.2(17d)SXB.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Examples

This example shows how to display a summary of bootflash information:

```
Router# show sup-bootflash

-#- ED --type-- --crc--- -seek-- nlen -length- -----date/time------ name
1   .. image    EBC8FC4D  A7487C    6 10700796 Nov 19 1999 07:07:37 halley
2   .. unknown  C7EB077D  EE2620   25  4644130 Nov 19 1999 07:50:44 cat6000-sup_
5-3-3-CSX.bin

645600 bytes available (15345184 bytes used)
Router#
```

This example shows how to display all bootflash information:

```
Router# show sup-bootflash all

-#- ED --type-- --crc--- -seek-- nlen -length- -----date/time------ name
1   .. image    EBC8FC4D  A7487C    6 10700796 Nov 19 1999 07:07:37 halley
2   .. unknown  C7EB077D  EE2620   25  4644130 Nov 19 1999 07:50:44 cat6000-sup_
5-3-3-CSX.bin

645600 bytes available (15345184 bytes used)

-------- F I L E   S Y S T E M   S T A T U S --------
  Device Number = 2
DEVICE INFO BLOCK: bootflash
  Magic Number          = 6887635   File System Vers = 10000    (1.0)
```

```
  Length                = 1000000   Sector Size      = 40000
  Programming Algorithm = 19        Erased State     = FFFFFFFF
  File System Offset    = 40000     Length = F40000
  MONLIB Offset         = 100       Length = F568
  Bad Sector Map Offset = 3FFF8     Length = 8
  Squeeze Log Offset    = F80000    Length = 40000
  Squeeze Buffer Offset = FC0000    Length = 40000
  Num Spare Sectors     = 0
    Spares:
STATUS INFO:
  Writable
  NO File Open for Write
  Complete Stats
  No Unrecovered Errors
  No Squeeze in progress
USAGE INFO:
  Bytes Used     = EA2620  Bytes Available = 9D9E0
  Bad Sectors    = 0       Spared Sectors  = 0
  OK Files       = 2       Bytes = EA2520
  Deleted Files  = 0       Bytes = 0
  Files w/Errors = 0       Bytes = 0

******** Intel SCS Status/Register Dump ********

COMMON MEMORY REGISTERS: Bank 0
  Intelligent ID Code  : 890089
  Compatible Status Reg: 800080

DEVICE TYPE:
  Layout                 : Paired x16 Mode
  Write Queue Size       : 64
  Queued Erase Supported : No

Router#
```

This example shows how to display information about the Flash chip:

```
Router# show sup-bootflash chips

******** Intel SCS Status/Register Dump ********

COMMON MEMORY REGISTERS: Bank 0
  Intelligent ID Code  : 890089
  Compatible Status Reg: 800080

DEVICE TYPE:
  Layout                 : Paired x16 Mode
  Write Queue Size       : 64
  Queued Erase Supported : No

Router#
```

This example shows how to display information about the file system:

```
Router# show sup-bootflash filesys

-------- F I L E   S Y S T E M   S T A T U S --------
  Device Number = 2
DEVICE INFO BLOCK: bootflash
  Magic Number          = 6887635   File System Vers = 10000    (1.0)
  Length                = 1000000   Sector Size      = 40000
  Programming Algorithm = 19        Erased State     = FFFFFFFF
  File System Offset    = 40000     Length = F40000
  MONLIB Offset         = 100       Length = F568
```

```
  Bad Sector Map Offset = 3FFF8     Length = 8
  Squeeze Log Offset    = F80000    Length = 40000
  Squeeze Buffer Offset = FC0000    Length = 40000
  Num Spare Sectors     = 0
    Spares:
STATUS INFO:
  Writable
  NO File Open for Write
  Complete Stats
  No Unrecovered Errors
  No Squeeze in progress
USAGE INFO:
  Bytes Used     = EA2620  Bytes Available = 9D9E0
  Bad Sectors    = 0       Spared Sectors  = 0
  OK Files       = 2       Bytes = EA2520
  Deleted Files  = 0       Bytes = 0
  Files w/Errors = 0       Bytes = 0

Router#
```

show system jumbomtu

To display the global maximum transmission unit (MTU) setting, use the **show system jumbomtu** command in privileged EXEC mode.

show system jumbomtu

Syntax Description

This command has no arguments or keywords.

Defaults

This command has no default settings.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(17d)SXB	Support for this command on the Supervisor Engine 2 was extended to Release 12.2(17d)SXB.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Examples

This example shows how to display the global MTU setting:

```
Router# show system jumbomtu

Global Ethernet MTU is 1550 bytes.
Router#
```

Related Commands

Command	Description
system jumbomtu	Sets the maximum size of the Layer 2 and Layer 3 packets.

show tech-support

To display general information about the router when it reports a problem, use the **show tech-support** command in privileged EXEC mode.

show tech-support [**page**] [**password**] [**cef | ipc | ipmulticast** [**vrf** *vrf-name*] **| isis | mpls | ospf** [*process-id* **| detail]** **| rsvp**]

Cisco 7600 Series

show tech-support [**cef | ipmulticast** [**vrf** *vrf-name*] **| isis | password** [**page**] **| platform | page | rsvp**]

Syntax Description

page	(Optional) Causes the output to display a page of information at a time.
password	(Optional) Leaves passwords and other security information in the output.
cef	(Optional) Displays **show** command output specific to Cisco Express Forwarding.
ipc	(Optional) Displays **show** command output specific to Inter-Process Communication (IPC).
ipmulticast	(Optional) Displays **show** command output related to the IP Multicast configuration, including Protocol Independent Multicast (PIM) information, Internet Group Management Protocol (IGMP) information, and Distance Vector Multicast Routing Protocol (DVMRP) information.
vrf *vrf-name*	(Optional) Specifies a multicast Virtual Private Network (VPN) routing and forwarding instance (VRF).
isis	(Optional) Displays **show** command output specific to Connectionless Network Service (CLNS) and Intermediate System-to-Intermediate System Protocol (IS-IS).
mpls	(Optional) Displays **show** command output specific to Multiprotocol Label Switching (MPLS) forwarding and applications.
ospf [*process-id* | **detail**]	(Optional) Displays **show** command output specific to Open Shortest Path First Protocol (OSPF) networking.
rsvp	(Optional) Displays **show** command output specific to Resource Reservation Protocol (RSVP) networking.
platform	(Optional) Displays platform-specific **show** command output.

Defaults

The output scrolls without page breaks.
Passwords and other security information are removed from the output.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
11.2	This command was introduced.
11.3(7), 11.2(16)	The output for this command was expanded to show additional information for **boot**, **bootflash**, **context**, and **traffic** for all enabled protocols.
12.0	The output for this command was expanded to show additional information for **boot**, **bootflash**, **context**, and **traffic** for all enabled protocols. The **cef**, **ipmulticast**, **isis**, **mlps**, and **ospf** keywords were added to this command.
12.2(13)T	Support for AppleTalk EIGRP, Apollo Domain, Banyan VINES, Novell Link-State Protocol, and XNS was removed from Cisco IOS software.
12.2(14)SX	Support for this command was added for the Supervisor Engine 720.
12.3(4)T	The output of this command was expanded to include the output from the **show inventory** command.
12.2(17d)SXB	Support for this command on the Supervisor Engine 2 was extended to Release 12.2(17d)SXB.
12.2(30)S	The **show tech-support ipmulticast** command was changed as follows: • Support for bidirectional PIM and Multicast VPN (MVPN) was added. • The **vrf** *vrf-name* option was added. The output of the **show tech-support ipmulticast** command (without the **vrf** *vrf-name* keyword and argument) was changed to include the output from these commands: • **show ip pim int df** • **show ip pim mdt** • **show ip pim mdt bgp** • **show ip pim rp metric**
12.3(16)	This command was integrated into Cisco IOS Release 12.3(16).
12.2(18)SXF	The **show tech-support ipmulticast** command was changed as follows: • Support for bidirectional PIM and MVPN was added. • The **vrf** *vrf-name* option was added. The output of the **show tech-support ipmulticast vrf** command was changed to include the output from these commands: • **show mls ip multicast rp-mapping gm-cache** • **show mmls gc process** • **show mmls msc rpdf-cache** The output of the **show tech-support ipmulticast** command (without the **vrf** *vrf-name* keyword and argument) was changed to include the output from these commands: • **show ip pim int df** • **show ip pim mdt** • **show ip pim mdt bgp** • **show ip pim rp metric** Support to interrupt and terminate the **show tech-support** output was added.

Release	Modification
12.4(4)T	This command was integrated into Cisco IOS Release 12.4(4)T.
12.4(7)	This command was integrated into Cisco IOS Release 12.4(7).
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.4(9)T	The output of this command was expanded to include partial **show dmvpn details** command output.

Usage Guidelines

To interrupt and terminate the **show tech-support** output, simultaneously press and release the **CTRL**, **ALT**, and **6** keys.

Press the **Return** key to display the next line of output, or press the **Spacebar** to display the next page of information. If you do not enter the **page** keyword, the output scrolls (that is, it does not stop for page breaks).

If you do not enter the **password** keyword, passwords and other security-sensitive information in the output are replaced with the label "<removed>."

The **show tech-support** command is useful for collecting a large amount of information about your routing device for troubleshooting purposes. The output of this command can be provided to technical support representatives when reporting a problem.

Note This command can generate a very large amount of output. You may want to redirect the output to a file using the **show inventory | redirect** *url* command syntax extension. Redirecting the output to a file also makes sending this output to your technical support representative easier. See the command documentation for **show <command> | redirect** for more information on this option.

The **show tech-support** command displays the output of a number of **show** commands at once. The output from this command varies depending on your platform and configuration. For example, access servers display voice-related **show** command output. Additionally, the **show** *protocol* **traffic** commands are displayed for only the protocols enabled on your device. For a sample display of the output of the **show tech-support** command, see the individual **show** command listed.

If you enter the **show tech-support** command without arguments, the output displays, but is not limited to, the equivalent of these **show** commands:

- **show appletalk traffic**
- **show bootflash**
- **show bootvar**
- **show buffers**
- **show cdp neighbors**
- **show cef**
- **show clns traffic**
- **show context**
- **show controllers**
- **show decnet traffic**
- **show disk0: all**
- show dmvpn details

- **show environment**
- show fabric channel-counters
- **show file systems**
- **show interfaces**
- **show interfaces switchport**
- **show interfaces trunk**
- **show ip interface**
- **show ip traffic**
- **show logging**
- **show mac-address-table**
- **show module**
- **show power**
- **show processes cpu**
- **show processes memory**
- **show running-config**
- **show spanning-tree**
- **show stacks**
- **show version**
- **show vlan**

Note Crypto information is not duplicated by the **show dmvpn details** command output.

Use of the optional **cef**, **ipc**, **ipmulticast**, **isis**, **mpls**, **ospf**, or **rsvp** keywords provides a way to display a number of **show** commands specific to a particular protocol or process in addition to the **show** commands listed previously.

For example, if your Technical Assistance Center (TAC) support representative suspects that you may have a problem in your Cisco Express Forwarding (CEF) configuration, you may be asked to provide the output of the **show tech-support cef** command. The **show tech-support** [**page**] [**password**] **cef** command will display the output from the following commands in addition to the output for the standard **show tech-support** command:

- **show adjacency summary**
- **show cef drop**
- **show cef events**
- **show cef interface**
- **show cef not-cef-switched**
- **show cef timers**
- **show interfaces stats**
- **show ip cef events summary**
- **show ip cef inconsistency records detail**
- **show ip cef summary**

If you enter the **ipmulticast** keyword, the output displays, but is not limited to, these **show** commands:

- **show ip dvmrp route**
- **show ip igmp groups**
- **show ip igmp interface**
- **show ip mcache**
- **show ip mroute**
- **show ip mroute count**
- **show ip pim interface**
- **show ip pim interface count**
- **show ip pim interface df**
- **show ip pim mdt**
- **show ip pim mdt bgp**
- **show ip pim neighbor**
- **show ip pim rp**
- **show ip pim rp metric**
- **show mls ip multicast rp-mapping gm-cache**
- show mmls gc process
- show mmls msc rpdf-cache

Examples

For a sample display of the output from the **show tech-support** command, refer to the documentation for the **show** commands listed in the "Usage Guidelines" section.

Related Commands

Command	Description
dir	Displays a list of files on a file system.
show appletalk traffic	Displays statistics about AppleTalk traffic, including MAC IP traffic.
show bootflash	Displays the contents of boot flash memory.
show bootvar	Displays the contents of the BOOT environment variable, the name of the configuration file pointed to by the CONFIG_FILE environment variable, the contents of the BOOTLDR environment variable, and the configuration register setting.
show buffers	Displays statistics for the buffer pools on the network server.
show cdp neighbors	Displays detailed information about neighboring devices discovered using Cisco Discovery Protocol.
show cef	Displays information about packets forwarded by Cisco Express Forwarding.
show clns traffic	Displays a list of the CLNS packets this router has seen.
show <*command***> | redirect**	Redirects the output of any **show** command to a file.
show context	Displays context data.
show controllers	Displays information that is specific to the hardware.

Command	Description
show controllers tech-support	Displays general information about a VIP card for problem reporting.
show decnet traffic	Displays the DECnet traffic statistics (including datagrams sent, received, and forwarded).
show disk:0	Displays flash or file system information for a disk located in slot 0:
show dmvpn details	Displays detail DMVPN information for each session, including Next Hop Server (NHS) and NHS status, crypto session information, and socket details.
show environment	Displays temperature, voltage, and blower information on the Cisco 7000 series routers, Cisco 7200 series routers, Cisco 7500 series routers, Cisco 7600 series routers, Cisco AS5300 series access servers, and the Gigabit Switch Router.
show fabric channel counters	Displays the fabric channel counters for a module.
show file system	Lists available file systems.
show interfaces	Displays statistics for all interfaces configured on the router or access server.
show interfaces switchport	Displays the administrative and operational status of a switching (nonrouting) port.
show interfaces trunk	Displays the interface-trunk information.
show inventory	Displays the product inventory listing and UDI of all Cisco products installed in the networking device.
show ip interface	Displays the usability status of interfaces configured for IP.
show ip traffic	Displays statistics about IP traffic.
show logging	Displays the state of syslog and the contents of the standard system logging buffer.
show mac-address table	Displays the MAC address table.
show module	Displays module status and information.
show power	Displays the current power status of system components.
show processes cpu	Displays information about the active processes.
show processes memory	Displays the amount of memory used.
show running-config	Displays the current configuration of your routing device.
show spanning-tree	Displays information about the spanning tree state.
show stacks	Displays the stack usage of processes and interrupt routines.
show version	Displays the configuration of the system hardware, the software version, the names and sources of configuration files, and the boot images.
show vlan	Displays VLAN information.

show usb controllers

To display USB host controller information, use the **show usb controllers** command in privileged EXEC mode.

show usb controllers [*controller-number*]

Syntax Description

controller-number	(Optional) Displays information only for the specified controller.

Defaults

Information about all controllers on the system are displayed.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.3(14)T	This command was introduced.
12.4(11)T	This command was integrated into the Cisco 7200VXR NPE-G2 platform.

Usage Guidelines

Use the **show usb controllers** command to display content such as controller register specific information, current asynchronous buffer addresses, and period scheduling information. You can also use this command to verify that copy operations are occurring successfully onto a USB flash module.

Examples

The following example is sample output from the **show usb controllers** command:

```
Router# show usb controllers

Name:1362HCD
Controller ID:1
Controller Specific Information:
    Revision:0x11
    Control:0x80
    Command Status:0x0
    Hardware Interrupt Status:0x24
    Hardware Interrupt Enable:0x80000040
    Hardware Interrupt Disable:0x80000040
    Frame Interval:0x27782EDF
    Frame Remaining:0x13C1
    Frame Number:0xDA4C
    LSThreshold:0x628
    RhDescriptorA:0x19000202
    RhDescriptorB:0x0
    RhStatus:0x0
    RhPort1Status:0x100103
    RhPort2Status:0x100303
    Hardware Configuration:0x3029
    DMA Configuration:0x0
    Transfer Counter:0x1
    Interrupt:0x9
```

```
    Interrupt Enable:0x196
    Chip ID:0x3630
    Buffer Status:0x0
    Direct Address Length:0x80A00
    ATL Buffer Size:0x600
    ATL Buffer Port:0x0
    ATL Block Size:0x100
    ATL PTD Skip Map:0xFFFFFFFF
    ATL PTD Last:0x20
    ATL Current Active PTD:0x0
    ATL Threshold Count:0x1
    ATL Threshold Timeout:0xFF

Int Level:1
Transfer Completion Codes:
         Success              :920            CRC             :0
         Bit Stuff            :0              Stall           :0
         No Response          :0              Overrun         :0
         Underrun             :0              Other           :0
         Buffer Overrun       :0              Buffer Underrun :0
Transfer Errors:
         Canceled Transfers   :2              Control Timeout :0
Transfer Failures:
         Interrupt Transfer   :0              Bulk Transfer   :0
         Isochronous Transfer :0              Control Transfer:0
Transfer Successes:
         Interrupt Transfer   :0              Bulk Transfer   :26
         Isochronous Transfer :0              Control Transfer:894

USBD Failures:
         Enumeration Failures :0              No Class Driver Found:0
         Power Budget Exceeded:0

USB MSCD SCSI Class Driver Counters:
         Good Status Failures :3              Command Fail    :0
         Good Status Timed out:0              Device not Found:0
         Device Never Opened  :0              Drive Init Fail :0
         Illegal App Handle   :0              Bad API Command :0
         Invalid Unit Number  :0              Invalid Argument:0
         Application Overflow :0              Device in use   :0
         Control Pipe Stall   :0              Malloc Error    :0
         Device Stalled       :0              Bad Command Code:0
         Device Detached      :0              Unknown Error   :0
         Invalid Logic Unit Num:0

USB Aladdin Token Driver Counters:
         Token Inserted       :1              Token Removed   :0
         Send Insert Msg Fail :0              Response Txns   :434
         Dev Entry Add Fail   :0              Request Txns    :434
         Dev Entry Remove Fail:0              Request Txn Fail:0
         Response Txn Fail    :0              Command Txn Fail:0
         Txn Invalid Dev Handle:0

USB Flash File System Counters:
         Flash Disconnected   :0              Flash Connected :1
         Flash Device Fail    :0              Flash Ok        :1
         Flash startstop Fail :0              Flash FS Fail   :0

USB Secure Token File System Counters:
         Token Inserted       :1              Token Detached  :0
         Token FS success     :1              Token FS Fail   :0
         Token Max Inserted   :0              Create Talker Failures:0
         Token Event          :0              Destroy Talker Failures:0
         Watched Boolean Create Failures:0
```

show usb device

To display USB device information, use the **show usb device** command in privileged EXEC mode.

show usb device [*controller-ID* [*device-address*]]

Syntax Description

controller-ID	(Optional) Displays information only for the devices under the specified controller.
device-address	(Optional) Displays information only for the device with the specified address.

Defaults

Information for all devices attached to the system are displayed.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.3(14)T	This command was introduced.
12.4(11)T	This command was integrated into the Cisco 7200VXR NPE-G2 platform.

Usage Guidelines

Use the **show usb device** command to display information for either a USB flash drive or a USB eToken, as appropriate.

Examples

The following example is sample output from the **show usb device** command:

```
Router# show usb device

Host Controller:1
Address:0x1
Device Configured:YES
Device Supported:YES
Description:DiskOnKey
Manufacturer:M-Sys
Version:2.0
Serial Number:0750D84030316868
Device Handle:0x1000000
USB Version Compliance:2.0
Class Code:0x0
Subclass Code:0x0
Protocol:0x0
Vendor ID:0x8EC
Product ID:0x15
Max. Packet Size of Endpoint Zero:64
Number of Configurations:1
Speed:Full
Selected Configuration:1
Selected Interface:0
```

```
Configuration:
    Number:1
    Number of Interfaces:1
    Description:
    Attributes:None
    Max Power:140 mA

    Interface:
        Number:0
        Description:
        Class Code:8
        Subclass:6
        Protocol:80
        Number of Endpoints:2

        Endpoint:
            Number:1
            Transfer Type:BULK
            Transfer Direction:Device to Host
            Max Packet:64
            Interval:0

        Endpoint:
            Number:2
            Transfer Type:BULK
            Transfer Direction:Host to Device
            Max Packet:64
            Interval:0

Host Controller:1
Address:0x11
Device Configured:YES
Device Supported:YES
Description:eToken Pro 4254
Manufacturer:AKS
Version:1.0
Serial Number:
Device Handle:0x1010000
USB Version Compliance:1.0
Class Code:0xFF
Subclass Code:0x0
Protocol:0x0
Vendor ID:0x529
Product ID:0x514
Max. Packet Size of Endpoint Zero:8
Number of Configurations:1
Speed:Low
Selected Configuration:1
Selected Interface:0

Configuration:
    Number:1
    Number of Interfaces:1
    Description:
    Attributes:None
    Max Power:60 mA

    Interface:
        Number:0
        Description:
        Class Code:255
        Subclass:0
        Protocol:0
        Number of Endpoints:0
```

Table 143 describes the significant fields shown in the display.

Table 143 show usb device Field Descriptions

Field	Description
Device handle	Internal memory handle allocated to the device.
Device Class code	The class code supported by the device. This number is allocated by the USB-IF. If this field is reset to 0, each interface within a configuration specifies its own class information, and the various interfaces operate independently. If this field is set to a value between 1 and FEH, the device supports different class specifications on different interfaces, and the interfaces may not operate independently. This value identifies the class definition used for the aggregate interfaces. If this field is set to FFH, the device class is vendor-specific.
Device Subclass code	The subclass code supported by the device. This number is allocated by the USB-IF.
Device Protocol	The protocol supported by the device. If this field is set to 0, the device does not use class-specific protocols on a device basis. If this field is set to 0xFF, the device uses a vendor-specific protocol on a device basis.
Interface Class code	The class code supported by the interface. If the value is set to 0xFF, the interface class is vendor specific. All other values are allocated by the USB-IF.
Interface Subclass code	The subclass code supported by the interface. All values are allocated by the USB-IF.
Interface Protocol	The protocol code supported by the interface. If this field is set to 0, the device does not use a class-specific protocol on this interface. If this field is set to 0xFF, the device uses a vendor-specific protocol for this interface.
Max Packet	Maximum data packet size, in bytes.

show usb driver

To display information about registered USB class drivers and vendor-specific drivers, use the **show usb driver** command in privileged EXEC mode.

show usb driver [*index*]

Syntax Description

index	(Optional) Displays information only for drivers on the specified index.

Defaults

Information about all drivers is displayed.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.3(14)T	This command was introduced.
12.4(11)T	This command was integrated into the Cisco 7200VXR NPE-G2 platform.

Examples

The following example is sample output for the **show usb driver** command:

```
Router# show usb driver

Index:0
Owner Mask:0x6
Class Code:0x0
Subclass Code:0x0
Protocol:0x0
Interface Class Code:0x8
Interface Subclass Code:0x6
Interface Protocol Code:0x50
Product ID:0x655BD598
Vendor ID:0x64E90000
Attached Devices:
    Controller ID:1, Device Address:1

Index:1
Owner Mask:0x1
Class Code:0x0
Subclass Code:0x0
Protocol:0x0
Interface Class Code:0x0
Interface Subclass Code:0x0
Interface Protocol Code:0x0
Product ID:0x514
Vendor ID:0x529
Attached Devices:
    Controller ID:1, Device Address:17

Index:2
Owner Mask:0x5
Class Code:0x9
```

```
Subclass Code:0x6249BD58
Protocol:0x2
Interface Class Code:0x5DC0
Interface Subclass Code:0x5
Interface Protocol Code:0xFFFFFFFF
Product ID:0x2
Vendor ID:0x1
Attached Devices:
    None

Index:3
Owner Mask:0x10
Class Code:0x0
Subclass Code:0x0
Protocol:0x0
Interface Class Code:0x0
Interface Subclass Code:0x0
Interface Protocol Code:0x0
Product ID:0x0
Vendor ID:0x0
Attached Devices:
    None
```

Table 144 describes the significant field shown in the display.

Table 144 show usb driver Field Descriptions

Field	Description
Owner Mask	Indicates the fields that are used in enumeration comparison. The driver can own different devices on the basis of their product or vendor IDs and device or interface class, subclass, and protocol codes.

show usb port

To sisplay USB root hub port information, use the **show usb port** command in privileged EXEC mode.

show usb port [*port-number*]

Syntax Description

port-number	(Optional) Displays information only for a specified. If the *port-number* is not issued, information for all root ports will be displayed.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.3(14)T	This command was introduced.

Examples

The following sample from the **show usb port** command shows the status of the port 1 on the router:

```
Router# show usb port

Port Number:0
Status:Enabled
Connection State:Connected
Speed:Full
Power State:ON

Port Number:1
Status:Enabled
Connection State:Connected
Speed:Low
Power State:ON
```

show usb tree

To display information about the port state and all attached devices, use the **show usb tree** command in privileged EXEC mode.

show usb tree

Syntax Description

This command has no arguments or keywords.

Command Modes

EXEC

Command History

Release	Modification
12.3(14)T	This command was introduced.

Examples

The following example is sample output from the **show usb tree** command. This output shows that both a USB flash module and a USB eToken are currently enabled.

```
Router# show usb tree

[Host Id:1, Host Type:1362HCD, Number of RH-Port:2]
<Root Port0:Power=ON      Current State=Enabled>
  Port0:(DiskOnKey) Addr:0x1 VID:0x08EC PID:0x0015 Configured (0x1000000)
<Root Port1:Power=ON      Current State=Enabled>
  Port1:(eToken Pro 4254) Addr:0x11 VID:0x0529 PID:0x0514 Configured (0x1010000)
```

show usbtoken

To display information about the USB eToken (such as the eToken ID), use the **show usbtoken** command in privileged EXEC mode.

show usbtoken[**0-9**]:[**all** | *filesystem*]

Syntax Description

0-9	(Optional) One of the ten available flash drives you can choose from; valid values: 0-9. If you do not specify a number, 0 is used by default
all	(Optional) All configuration files stored on the eToken.
filesystem	(Optional) Name of a configuration file.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.3(14)T	This command was introduced.
12.4(11)T	This command was integrated into the Cisco 7200VXR NPE-G2 platform.

Usage Guidelines

Use the **show usbtoken** command to verify whether a USB eToken is inserted in the router.

Examples

The following example is sample output from the **show usbtoken** command:

```
Router# show usbtoken0

Token ID           :43353334
Token device name  : token0
Vendor name        : Vendor34
Product Name       :Etoken Pro
Serial number      : 22273a334353
Firmware version   :   4.1.3.2
Total memory size  : 32 KB
Free memory size   : 16 KB
FIPS version       :  Yes/No
Token state        :  "Active" | "User locked" | "Admin locked" | "System Error" |
"Uknown"
ATR (Answer To Reset) :"3B F2 98 0  FF C1 10 31 FE 55 C8 3"
```

Table 145 describes the significant fields shown in the display.

Table 145 show usbtoken Field Descriptions

Field	Description
Token ID	Token identifier.

Table 145 show usbtoken Field Descriptions (continued)

Field	Description
Token device name	A unique name derived by the token driver.
ATR (Answer to Reset)	Information replied by Smart cards when a reset command is issued.

show version

To display information about the currently loaded software along with hardware and device information, use the **show version** command in user EXEC, privileged EXEC, or diagnostic mode.

show version

Cisco ASR 1000 Series Routers

show version [*rp-slot*] [**installed** [**user-interface**] | **provisioned** | **running**]

Syntax Description

rp-slot	Specifies the software of the RP in a specific RP slot of a Cisco ASR 1000 Series Router. Options include: • **r0**—the RP in RP slot 0. • **r1**—the RP in RP slot 1. • **rp active**—the active RP. • **rp standby**—the standby RP.
installed	Specifies information on the software installed on the RP
user-interface	Specifies information on the files related to the user-interface.
provisioned	Specifies information on the software files that are provisioned.
running	Specifies information on the files currently running.

Defaults

No default behavior or values.

Command Modes

User EXEC (>)
Privileged EXEC (#)
Diagnostic (diag)—Cisco ASR 1000 Series Routers only

Command History

Release	Modification
9.0	This command was introduced.
12.1EC	This command was integrated into Cisco IOS Release 12.1EC.
12.1(1a)T1	This command was modified to include information about the clock card on CMTS routers.
12.3BC	This command was integrated into Cisco IOS Release 12.3BC.
12.3(4)T	The output format of this command was updated.
12.2(14)SX	Support for this command was introduced on the Supervisor Engine 720.
12.2(17d)SXB	Support for this command on the Supervisor Engine 2 was extended to 12.2(17d)SXB.
12.2(25)S	The output format of this command was updated.
12.2(33)SCA	This command was integrated into Cisco IOS Release 12.2(33)SCA. Support for the Cisco uBR7225VXR router was added.

Release	Modification
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
Cisco IOS XE Release 2.1	This command was introduced on the Cisco ASR 1000 Series Routers, and the following enhancements were introduced: • the command became available in diagnostic mode. • the *rp-slot*, **installed**, **user-interface**, **provisioned**, and **running** options all became available for the first time.

Usage Guidelines

This command displays information about the Cisco IOS software version currently running on a routing device, the ROM Monitor and Bootflash software versions, and information about the hardware configuration, including the amount of system memory. Because this command displays both software and hardware information, the output of this command is the same as the output of the **show hardware** command. (The **show hardware** command is a command alias for the **show version** command.)

Specifically, the **show version** command provides the following information:

- Software information
  - Main Cisco IOS image version
  - Main Cisco IOS image capabilities (feature set)
  - Location and name of bootfile in ROM
  - Bootflash image version (depending on platform)
- Device-specific information
  - Device name
  - System uptime
  - System reload reason
  - Config-register setting
  - Config-register settings for after the next reload (depending on platform)
- Hardware information
  - Platform type
  - Processor type
  - Processor hardware revision
  - Amount of main (processor) memory installed
  - Amount I/O memory installed
  - Amount of Flash memory installed on different types (depending on platform)
  - Processor board ID

The output of this command uses the following format:

```
Cisco IOS Software, <platform> Software (<image-id>), Version <software-version>,
<software-type>
Technical Support: http://www.cisco.com/techsupport
Copyright (c) <date-range> by Cisco Systems, Inc.
Compiled <day> <date> <time> by <compiler-id>

ROM: System Bootstrap, Version <software-version>,  <software-type>
BOOTLDR: <platform> Software (image-id), Version <software-version>,  <software-type>
```

```
<router-name> uptime is <w> weeks, <d> days, <h> hours, <m> minutes
System returned to ROM by reload at <time> <day> <date>
System image file is "<filesystem-location>/<software-image-name>"
Last reload reason: <reload-reason>

Cisco <platform-processor-type> processor (revision <processor-revision-id>) with
<free-DRAM-memory>K/<packet-memory>K bytes of memory.
Processor board ID <ID-number>
<CPU-type> CPU at <clock-speed>Mhz, Implementation <number>, Rev <Revision-number>,
<kilobytes-Processor-Cache-Memory>KB <cache-Level> Cache
```

See the Examples section for descriptions of the fields in this output.

Cisco ASR 1000 Series Routers

Entering **show version** without any of the options on the Cisco ASR 1000 Series Router will generate output similar to **show version** on other Cisco routers.

In order to understand the **show version** output on Cisco ASR 1000 Series Routers, it is important to understand that the individual sub-packages run the processes on the router. Among other things, the output of this command provides information on where various individual sub-packages are stored on the router, and which processes these individual sub-packages are and are not currently running.

More specifically, the **show version installed** command displays each individual sub-package file on the router, the hardware where the sub-package could be running, and whether the sub-package is currently being run on that hardware.

The **show version provisioned** command displays only the individual sub-packages that can be provisioned, which are the RP-specific sub-packages (RP Access, RP Base, RP Control, and RP IOS) and the provisioning file. The output includes the individual sub-package file, the hardware where the sub-package could be running, and whether the sub-package is currently being run on that hardware.

The **show version running** command displays only the individual sub-packages that are currently active. The output includes the individual sub-package file and the hardware where the sub-package is running.

Examples

Cisco 3660 Router

The following is sample output from the **show version** command issued on a Cisco 3660 running Cisco IOS Release 12.3(4)T:

```
Router# show version

Cisco IOS Software, 3600 Software (C3660-I-M), Version 12.3(4)T
TAC Support: http://www.cisco.com/tac
Copyright (c) 1986-2003 by Cisco Systems, Inc.
Compiled Thu 18-Sep-03 15:37 by ccai

ROM: System Bootstrap, Version 12.0(6r)T, RELEASE SOFTWARE (fc1)
ROM:

C3660-1 uptime is 1 week, 3 days, 6 hours, 41 minutes
System returned to ROM by power-on
System image file is "slot0:tftpboot/c3660-i-mz.123-4.T"

Cisco 3660 (R527x) processor (revision 1.0) with 57344K/8192K bytes of memory.
Processor board ID JAB055180FF
R527x CPU at 225Mhz, Implementation 40, Rev 10.0, 2048KB L2 Cache


3660 Chassis type: ENTERPRISE
```

```
2 FastEthernet interfaces
4 Serial interfaces
DRAM configuration is 64 bits wide with parity disabled.
125K bytes of NVRAM.
16384K bytes of processor board System flash (Read/Write)

Flash card inserted. Reading filesystem...done.
20480K bytes of processor board PCMCIA Slot0 flash (Read/Write)

Configuration register is 0x2102
```

Cisco 7200 Router

The following is sample output from the **show version** command issued on a Cisco 7200 router running Cisco IOS Release 12.4(4)T. This output shows the total bandwidth capacity and the bandwith capacity that is configured on the Cisco 7200. Displaying bandwidth capacity is available in Cisco IOS Release 12.2 and later releases.

```
Router# show version

Cisco IOS Software, 7200 Software (C7200-JS-M), Version 12.4(4)T, RELEASE SOFTW)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2005 by Cisco Systems, Inc.
Compiled Thu 27-Oct-05 05:58 by ccai

ROM: System Bootstrap, Version 12.1(20000710:044039) [nlaw-121E_npeb 117], DEVEE
BOOTLDR: 7200 Software (C7200-KBOOT-M), Version 12.3(16), RELEASE SOFTWARE (fc4)

router uptime is 5 days, 18 hours, 2 minutes
System returned to ROM by reload at 02:45:12 UTC Tue Feb 14 2006
System image file is "disk0:c7200-js-mz.124-4.T"
Last reload reason: Reload Command


Cisco 7206VXR (NPE400) processor (revision A) with 491520K/32768K bytes of memo.
Processor board ID 26793934
R7000 CPU at 350MHz, Implementation 39, Rev 3.2, 256KB L2 Cache
6 slot VXR midplane, Version 2.6

Last reset from power-on

PCI bus mb0_mb1 (Slots 0, 1, 3 and 5) has a capacity of 600 bandwidth points.
Current configuration on bus mb0_mb1 has a total of 440 bandwidth points.
This configuration is within the PCI bus capacity and is supported.

PCI bus mb2 (Slots 2, 4, 6) has a capacity of 600 bandwidth points.
Current configuration on bus mb2 has a total of 390 bandwidth points
This configuration is within the PCI bus capacity and is supported.

Please refer to the following document "Cisco 7200 Series Port Adaptor
Hardware Configuration Guidelines" on Cisco.com <http://www.cisco.com>
for c7200 bandwidth points oversubscription and usage guidelines.


4 Ethernet interfaces
2 FastEthernet interfaces
2 ATM interfaces
125K bytes of NVRAM.

62976K bytes of ATA PCMCIA card at slot 0 (Sector size 512 bytes).
125952K bytes of ATA PCMCIA card at slot 1 (Sector size 512 bytes).
8192K bytes of Flash internal SIMM (Sector size 256K).
Configuration register is 0x2002
```

```
Router#
```

For information about PCI buses and bandwidth calculation, go to http://www.cisco.com/univercd/cc/td/doc/product/core/7206/port_adp/config/3875in.htm#wp1057192.

Table 146 describes the significant fields shown in the display.

Table 146 show version Field Descriptions

Field	Description
Cisco IOS Software, *platform* Software (*image-id*), Version *software-version*, *release-type* For example: Cisco IOS Software, 7200 Software (C7200-G4JS-M), Version 12.3(4)T	*platform*—Cisco hardware device name. *image-id*—The coded software image identifier, in the format *platform-features-format* (for example, "c7200-g4js-mz". *software-version*—The Cisco IOS software release number, in the format *x.y*(*z*)*A*, where *x.y* is the main release identifier, *z* is the maintenance release number, and *A*, where applicable, is the special release train identifier. For example, 12.3(4)T indicates the fourth maintenance release of the 12.3T special technology release train. **Note** In the full software image filename, 12.3(4)T appears as 123-4.T. In the IOS Upgrade Planner, 12.3(4)T appears as 12.3.4T (ED). *release-type*—The description of the release type. Possible values include MAINTENANCE [for example, 12.3(3)] or INTERIM [for example, 12.3(3.2)]. **Tip** Refer to "The ABC's of Cisco IOS Networking" (available on Cisco.com) for more information on Cisco IOS software release numbering and software versions. Cisco IOS is a registered trademark (R) of Cisco Systems, Inc.
Technical Support: http://www.cisco.com/techsupport Copyright (c) *date-range* by Cisco Systems, Inc.	The Cisco Technical Support & Documentation website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content. Cisco IOS software, including the source code, user-help, and documentation, is copyrighted by Cisco Systems, Inc. It is Cisco's policy to enforce its copyrights against any third party who infringes on its copyright.
ROM: System Bootstrap, Version 12.0(6r)T, RELEASE SOFTWARE (fc1)	The system "bootstrap" software, stored in ROM memory.
BOOTFLASH:	The system "bootflash" software, stored in Flash memory (if applicable).

Table 146 show version Field Descriptions (continued)

Field	Description
device uptime is ... For example: C3660-1 uptime is 1 week, 3 days, 6 hours, 41 minutes	The amount of time the system has been up and running.
System returned to ROM by *reload-reason* at *time day date* For example: System returned to ROM by reload at 20:56:53 UTC Tue Nov 4 2003	Shows the last recorded reason for a system reload, and time of last reload.
Last reload reason: *reload-reason* For example: Last reload reason: Reload command	Shows the last recorded reason for a system reload.
Last reset from *reset-reason* For example: Last reset from power-on	Shows the last recorded reason for a system reset. Possible *reset-reason* values include: • power-on—System was reset with the initial power on or a power cycling of the device. • s/w peripheral—System was reset due to a software peripheral. • s/w nmi—System was reset by a nonmaskable interrupt (NMI) originating in the system software. For example, on some systems, you can configure the device to reset automatically if two or more fans fail. • push-button—System was reset by manual activation of a RESET push-button (also called a hardware NMI). • watchdog—System was reset due to a watchdog process. • unexpected value—May indicate a bus error, such as for an attempt to access a nonexistent address (for example, “System restarted by bus error at PC 0xC4CA, address 0x210C0C0”). (This field was formerly labeled as the “System restarted by” field.”)
System image file is “*file-location/file-name*” For example: System image file is "slot0:tftpboot/c3660-i-mz.123-3.9.T2"	Displays the file location (local or remote filesystem) and the system image name.

Table 146 show version Field Descriptions (continued)

Field	Description
Cisco *platform* (*processor-type*) processor (revision *processor-revision-id*) with *free-DRAM-memory* K/ *packet-memory* K bytes of memory. Example—Separate DRAM and Packet Memory: Cisco RSP4 (R5000) processor with 65536K/2072K bytes of memory Example—Combined DRAM and Packet Memory: Cisco 3660 (R527x) processor (revision 1.0) with 57344K/8192K bytes of memory.	This line can be used to determine how much Dynamic RAM (DRAM) is installed on your system, in order to determine if you meet the "Min. Memory" requirement for a software image. DRAM (including SDRAM) is used for system processing memory and for packet memory. Two values, separated by a slash, are given for DRAM: The first value tells you how DRAM is available for system processing, and the second value tells you how much DRAM is being used for Packet memory. The first value, Main Processor memory, is either: • The amount of DRAM available for the processor, or • The total amount of DRAM installed on the system. The second value, Packet memory, is either: • The total physical input/output (I/O) memory (or "Fast memory") installed on the router (Cisco 4000, 4500, 4700, and 7500 series), or • The amount of "shared memory" used for packet buffering. In the shared memory scheme (Cisco 2500, 2600, 3600, and 7200 Series), a percentage of DRAM is used for packet buffering by the router's network interfaces. **Note** The terms "I/O memory" or "iomem"; "shared memory"; "Fast memory" and "PCI memory" all refer to "Packet Memory". Packet memory is either separate physical RAM or shared DRAM. **Separate DRAM and Packet Memory** The 4000, 4500, 4700, and 7500 series routers have separate DRAM and Packet memory, so you only need to look at the first number to determine total DRAM. In the example to the left for the Cisco RSP4, the first value shows that the router has 65536K (65,536 kilobytes, or 64 megabytes) of DRAM. The second value, 8192K, is the Packet memory. **Combined DRAM and Packet Memory** The 2500, 2600, 3600, and 7200 series routers require a minimum amount of I/O memory to support certain interface processors. The 1600, 2500, 2600, 3600, and 7200 series routers use a fraction of DRAM as Packet memory, so you need to add both numbers to find out the real amount of DRAM. In the example to the left for the Cisco 3660, the router has 57,344 kilobytes (KB) of free DRAM and 8,192 KB dedicated to Packet memory. Adding the two numbers together gives you 57,344K + 8,192K = 65,536K, or 64 megabytes (MB) of DRAM.

Table 146 show version Field Descriptions (continued)

Field	Description
	For more details on memory requirements, see the document "How to Choose a Cisco IOS® Software Release" on Cisco.com.
Configuration register is *value* For example: Configuration register is 0x2142 (will be 0x2102 at next reload)	Shows the current configured hex value of the software configuration register. If the value has been changed with the **config-register** command, the register value that will be used at the next reload is displayed in parenthesis. The boot field (final digit) of the software configuration register dictates what the system will do after a reset. For example, when the boot field of the software configuration register is set to 00 (for example, 0x0), and you press the NMI button on a Performance Route Processor (PRP), the user-interface remains at the ROM monitor prompt (rommon>) and waits for a user command to boot the system manually. But if the boot field is set to 01 (for example, 0x1), the system automatically boots the first Cisco IOS image found in the onboard Flash memory SIMM on the PRP. The factory-default setting for the configuration register is 0x2102. This value indicates that the router will attempt to load a Cisco IOS software image from Flash memory and load the startup configuration file.

Catalyst 6500 Series Switches and Cisco 7600 Series Routers

This example shows how to display the configuration of the system hardware, the software version, the names and sources of configuration files, and the boot images:

```
Router# show version
Cisco Internetwork Operating System Software
IOS (tm) c6sup2_rp Software (c6sup2_rp-JSV-M), Version 12.1 (nightly.E020626) NIG
HTLY BUILD
Copyright (c) 1986-2002 by cisco Systems, Inc.
Compiled Wed 26-Jun-02 06:20 by
Image text-base: 0x40008BF0, data-base: 0x419BA000

ROM: System Bootstrap, Version 12.1(11r)E1, RELEASE SOFTWARE (fc1)

Router uptime is 2 weeks, 8 hours, 48 minutes
Time since Router switched to active is 1 minute
System returned to ROM by power-on (SP by power-on)
System image file is "sup-bootflash:c6sup22-jsv-mz"

cisco Catalyst 6000 (R7000) processor with 112640K/18432K bytes of memory.
Processor board ID SAD06210067
R7000 CPU at 300Mhz, Implementation 39, Rev 3.3, 256KB L2, 1024KB L3 Cache
Last reset from power-on
Bridging software.
X.25 software, Version 3.0.0.
SuperLAT software (copyright 1990 by Meridian Technology Corp).
TN3270 Emulation software.
3 Virtual Ethernet/IEEE 802.3  interface(s)
48 FastEthernet/IEEE 802.3 interface(s)
381K bytes of non-volatile configuration memory.

16384K bytes of Flash internal SIMM (Sector size 512K).
```

```
Configuration register is 0x2102
Router#
```

Table 147 describes the fields that are shown in the example.

Table 147 show version Field Descriptions

Field	Description
IOS (tm) c6sup2_rp Software (c6sup2_rp-JSV-M), Version 12.1(nightly.E020626) NIGHTLY BUILD	Version number. Always specify the complete version number when reporting a possible software problem. In the example output, the version number is 12.1.
ROM: System Bootstrap, Version 12.1(11r)E1, RELEASE SOFTWARE (fc1)	Bootstrap version string.
BOOTFLASH: 7200 Software (C7200-BOOT-M), Version 11.1(472), RELEASE SOFTWARE	Boot version string.
Router uptime is	Amount of time that the system has been up and running.
Time since Router switched to active	Amount of time since switchover occurred.
System restarted by	Log of how the system was last booted, both as a result of normal system startup and of system error. For example, information can be displayed to indicate a bus error that is typically the result of an attempt to access a nonexistent address, as follows: System restarted by bus error at PC 0xC4CA, address 0x210C0C0
System image file is	If the software was booted over the network, the Internet address of the boot host is shown. If the software was loaded from onboard ROM, this line reads “running default software.”
cisco Catalyst 6000 (R7000) processor with 112640K/18432K bytes of memory.	Remaining output in each display that shows the hardware configuration and any nonstandard software options.
Configuration register is	Configuration register contents that are displayed in hexadecimal notation.

The output of the **show version** EXEC command can provide certain messages, such as bus error messages. If such error messages appear, report the complete text of this message to your technical support specialist.

Cisco uBR7246VXR Router

The following is sample output from the **show version** command for a Cisco uBR7246 VXR with the cable clock card installed:

```
Router# show version

Cisco Internetwork Operating System Software
IOS (tm) 7200 Software (UBR7200-P-M), Version 12.1(10)EC, RELEASE SOFTWARE
TAC Support: http://www.cisco.com/tac
```

```
Copyright (c) 1986-2000 by cisco Systems, Inc.
Compiled Wed 02-Feb-00 16:49 by ccai
Image text-base:0x60008900, data-base:0x61192000

ROM:System Bootstrap, Version 12.0(15)SC, RELEASE SOFTWARE

VXR1 uptime is 2 days, 1 hour, 24 minutes
System returned to ROM by power-on at 10:54:38 PST Sat Feb 5 2000
System restarted at 11:01:08 PST Sat Feb 5 2000
System image file is "slot1:ubr7200-p-mz.121-0.8.T"

cisco uBR7246VXR (NPE300) processor (revision B) with 122880K/40960K bytes of memory.
Processor board ID SAB0329005N
R7000 CPU at 262Mhz, Implementation 39, Rev 1.0, 256KB L2, 2048KB L3 Cache
6 slot VXR midplane, Version 2.0

Last reset from power-on
X.25 software, Version 3.0.0.
National clock card with T1 controller
1 FastEthernet/IEEE 802.3 interface(s)
2 Cable Modem network interface(s)
125K bytes of non-volatile configuration memory.

16384K bytes of Flash PCMCIA card at slot 0 (Sector size 128K).
20480K bytes of Flash PCMCIA card at slot 1 (Sector size 128K).
4096K bytes of Flash internal SIMM (Sector size 256K).
Configuration register is 0x0

Router#
```

Table 148 describes significant fields shown in these displays.

Table 148 show version Field Descriptions

Field	Description
IOS (tm) 7200 Software (UBR7200-P-M), Version xx.x	Always specify the complete version number when reporting a possible software problem. In the example, the version number is Cisco IOS Release 12.1(10)EC.
ROM: System Bootstrap	Bootstrap version string.
Router uptime is	The amount of time the system has been up and running.
System restarted at	Also displayed is a log of how the system was last booted, as a result of normal system startup or system error.
System image file is	If the software was booted over the network, the Internet address of the boot host is shown. If the software was loaded from onboard ROM, this line reads “running default software.”
cisco uBR7246VXR (NPE300) processor	The remaining output in each display shows the hardware configuration and any nonstandard software options.
Configuration register is	The configuration register contents, displayed in hexadecimal notation.

The output of the **show version** command can also provide certain messages, such as bus error messages. If such error messages appear, report the complete text of this message to your technical support specialist.

Cisco uBR10012 Router

The following example shows sample output from the show version command on a Cisco uBR10012 universal broadband router running Cisco IOS Release 12.3(17b)BC4:

```
Router> show version
Cisco Internetwork Operating System Software
IOS (tm) 10000 Software (UBR10K2-K9P6U2-M), Version 12.3(17b)BC4, RELEASE SOFTWA
RE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2006 by cisco Systems, Inc.
Compiled Wed 22-Nov-06 11:41 by tinhuang
Image text-base: 0x60010F0C, data-base: 0x62480000

ROM: System Bootstrap, Version 12.0(20020314:211744) [REL-pulsar_sx.ios-rommon 1
12], DEVELOPMENT SOFTWARE

ubr10k uptime is 2 days, 22 hours, 13 minutes
System returned to ROM by reload at 01:34:58 UTC Sun Jun 8 2008
System image file is "disk0:ubr10k2-k9p6u2-mz.123-17b.BC4"
Last reload reason: Reload command



This product contains cryptographic features and is subject to United
States and local country laws governing import, export, transfer and
use. Delivery of Cisco cryptographic products does not imply
third-party authority to import, export, distribute or use encryption.
Importers, exporters, distributors and users are responsible for
compliance with U.S. and local country laws. By using this product you
agree to comply with applicable laws and regulations. If you are unable
to comply with U.S. and local laws, return this product immediately.

A summary of U.S. laws governing Cisco cryptographic products may be found at:
http://www.cisco.com/wwl/export/crypto/tool/stqrg.html

If you require further assistance please contact us by sending email to
export@cisco.com.

cisco uBR10000 (PRE2-RP) processor with 946175K/98304K bytes of memory.
Processor board ID TBA05380380
R7000 CPU at 500MHz, Implementation 39, Rev 4.1, 256KB L2, 8192KB L3 Cache
Backplane version 1.1, 8 slot

Last reset from register reset
PXF processor tmc0 is running.
PXF processor tmc1 is running.
PXF processor tmc2 is running.
PXF processor tmc3 is running.
1 TCCplus card(s)
1 FastEthernet/IEEE 802.3 interface(s)
3 Gigabit Ethernet/IEEE 802.3 interface(s)
24 Cable Modem network interface(s)
2045K bytes of non-volatile configuration memory.

125440K bytes of ATA PCMCIA card at slot 0 (Sector size 512 bytes).
125440K bytes of ATA PCMCIA card at slot 1 (Sector size 512 bytes).
65536K bytes of Flash internal SIMM (Sector size 512KB).
Secondary is up.
Secondary has 1044480K bytes of memory.

Configuration register is 0x2102
```

Cisco ASR 1000 Series Routers

In the following example, the **show version installed** command is entered on a Cisco ASR 1000 Series Router in diagnostic mode. Note that the output shows what every file that can be found in the consolidated package is or is not currently running (provisioning file, RP Access, RP Base, RP Control, RP IOS, ESP Base, SIP Base, SIP SPA).

```
Router#show version installed
Package: Provisioning File, version: n/a, status: active
  File: bootflash:packages.conf, on: RP0
  Built: n/a, by: n/a
  File SHA1 checksum: 0b9f2c7c3d81d8455a918f285c078463c04a0cab

Package: rpbase, version: v122_33_xn_asr_rls0_throttle, status: active
  File: bootflash:asr1000rp1-rpbase.v122_33_xn_asr_rls0_throttle.pkg, on: RP0
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: 193c4810becc2a6097645f0b68f5684004bd3ab3

Package: rpaccess-k9, version: v122_33_xn_asr_rls0_throttle, status: active
  File: bootflash:asr1000rp1-rpaccess-k9.v122_33_xn_asr_rls0_throttle.pkg, on: RP0
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: 328c3d1e10f006304ce9543ab68e914b43c41b1e

Package: rpcontrol, version: v122_33_xn_asr_rls0_throttle, status: active
  File: bootflash:asr1000rp1-rpcontrol.v122_33_xn_asr_rls0_throttle.pkg, on: RP0/0
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: e4152b7fe3c2b8aca07ce1e8ad6d5a54d6d20689

Package: rpios-advipservicesk9, version: v122_33_xn_asr_rls0_throttle, status: active
  File: bootflash:asr1000rp1-rpios-advipservicesk9.v122_33_xn_asr_rls0_throttle.pkg, on:
RP0/0
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: 7f7f87f2c198c38e7b58214478c5b28ee3c7b567

Package: rpcontrol, version: v122_33_xn_asr_rls0_throttle, status: inactive
  File: bootflash:asr1000rp1-rpcontrol.v122_33_xn_asr_rls0_throttle.pkg, on: RP0/1
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: e4152b7fe3c2b8aca07ce1e8ad6d5a54d6d20689

Package: rpios-advipservicesk9, version: v122_33_xn_asr_rls0_throttle, status: inactive
  File: bootflash:asr1000rp1-rpios-advipservicesk9.v122_33_xn_asr_rls0_throttle.pkg, on:
RP0/1
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: 7f7f87f2c198c38e7b58214478c5b28ee3c7b567

Package: rpbase, version: v122_33_xn_asr_rls0_throttle, status: inactive
  File: bootflash:asr1000rp1-rpbase.v122_33_xn_asr_rls0_throttle.pkg, on: RP1
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: 193c4810becc2a6097645f0b68f5684004bd3ab3

Package: rpaccess-k9, version: v122_33_xn_asr_rls0_throttle, status: inactive
  File: bootflash:asr1000rp1-rpaccess-k9.v122_33_xn_asr_rls0_throttle.pkg, on: RP1
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: 328c3d1e10f006304ce9543ab68e914b43c41b1e

Package: rpcontrol, version: v122_33_xn_asr_rls0_throttle, status: inactive
  File: bootflash:asr1000rp1-rpcontrol.v122_33_xn_asr_rls0_throttle.pkg, on: RP1/0
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: e4152b7fe3c2b8aca07ce1e8ad6d5a54d6d20689

Package: rpios-advipservicesk9, version: v122_33_xn_asr_rls0_throttle, status: inactive
  File: bootflash:asr1000rp1-rpios-advipservicesk9.v122_33_xn_asr_rls0_throttle.pkg, on:
RP1/0
  Built: 2007-11-11_17.16, by: mcpre
```

```
  File SHA1 checksum: 7f7f87f2c198c38e7b58214478c5b28ee3c7b567

Package: rpcontrol, version: v122_33_xn_asr_rls0_throttle, status: inactive
  File: bootflash:asr1000rp1-rpcontrol.v122_33_xn_asr_rls0_throttle.pkg, on: RP1/1
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: e4152b7fe3c2b8aca07ce1e8ad6d5a54d6d20689

Package: rpios-advipservicesk9, version: v122_33_xn_asr_rls0_throttle, status: inactive
  File: bootflash:asr1000rp1-rpios-advipservicesk9.v122_33_xn_asr_rls0_throttle.pkg, on:
RP1/1
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: 7f7f87f2c198c38e7b58214478c5b28ee3c7b567

Package: espbase, version: v122_33_xn_asr_rls0_throttle, status: active
  File: bootflash:asr1000rp1-espbase.v122_33_xn_asr_rls0_throttle.pkg, on: FP0
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: b1c004ed151cf60f0ce250f6ea710f43707fb010

Package: espbase, version: v122_33_xn_asr_rls0_throttle, status: inactive
  File: bootflash:asr1000rp1-espbase.v122_33_xn_asr_rls0_throttle.pkg, on: FP1
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: b1c004ed151cf60f0ce250f6ea710f43707fb010

Package: sipbase, version: v122_33_xn_asr_rls0_throttle, status: active
  File: bootflash:asr1000rp1-sipbase.v122_33_xn_asr_rls0_throttle.pkg, on: CC0
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: bd34a8a23d001f9cefcac8853a31b62ffd8272a4

Package: sipspa, version: v122_33_xn_asr_rls0_throttle, status: active
  File: bootflash:asr1000rp1-sipspa.v122_33_xn_asr_rls0_throttle.pkg, on: CC0/0
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: 6ad199569dad7d8b35beac2c8a72b080f9662897

Package: sipspa, version: v122_33_xn_asr_rls0_throttle, status: active
  File: bootflash:asr1000rp1-sipspa.v122_33_xn_asr_rls0_throttle.pkg, on: CC0/1
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: 6ad199569dad7d8b35beac2c8a72b080f9662897

Package: sipspa, version: v122_33_xn_asr_rls0_throttle, status: active
  File: bootflash:asr1000rp1-sipspa.v122_33_xn_asr_rls0_throttle.pkg, on: CC0/2
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: 6ad199569dad7d8b35beac2c8a72b080f9662897

Package: sipspa, version: v122_33_xn_asr_rls0_throttle, status: inactive
  File: bootflash:asr1000rp1-sipspa.v122_33_xn_asr_rls0_throttle.pkg, on: CC0/3
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: 6ad199569dad7d8b35beac2c8a72b080f9662897

Package: sipbase, version: v122_33_xn_asr_rls0_throttle, status: active
  File: bootflash:asr1000rp1-sipbase.v122_33_xn_asr_rls0_throttle.pkg, on: CC1
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: bd34a8a23d001f9cefcac8853a31b62ffd8272a4

Package: sipspa, version: v122_33_xn_asr_rls0_throttle, status: active
  File: bootflash:asr1000rp1-sipspa.v122_33_xn_asr_rls0_throttle.pkg, on: CC1/0
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: 6ad199569dad7d8b35beac2c8a72b080f9662897

Package: sipspa, version: v122_33_xn_asr_rls0_throttle, status: active
  File: bootflash:asr1000rp1-sipspa.v122_33_xn_asr_rls0_throttle.pkg, on: CC1/1
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: 6ad199569dad7d8b35beac2c8a72b080f9662897

Package: sipspa, version: v122_33_xn_asr_rls0_throttle, status: active
```

```
  File: bootflash:asr1000rp1-sipspa.v122_33_xn_asr_rls0_throttle.pkg, on: CC1/2
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: 6ad199569dad7d8b35beac2c8a72b080f9662897

Package: sipspa, version: v122_33_xn_asr_rls0_throttle, status: inactive
  File: bootflash:asr1000rp1-sipspa.v122_33_xn_asr_rls0_throttle.pkg, on: CC1/3
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: 6ad199569dad7d8b35beac2c8a72b080f9662897

Package: sipbase, version: v122_33_xn_asr_rls0_throttle, status: inactive
  File: bootflash:asr1000rp1-sipbase.v122_33_xn_asr_rls0_throttle.pkg, on: CC2
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: bd34a8a23d001f9cefcac8853a31b62ffd8272a4

Package: sipspa, version: v122_33_xn_asr_rls0_throttle, status: inactive
  File: bootflash:asr1000rp1-sipspa.v122_33_xn_asr_rls0_throttle.pkg, on: CC2/0
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: 6ad199569dad7d8b35beac2c8a72b080f9662897

Package: sipspa, version: v122_33_xn_asr_rls0_throttle, status: inactive
  File: bootflash:asr1000rp1-sipspa.v122_33_xn_asr_rls0_throttle.pkg, on: CC2/1
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: 6ad199569dad7d8b35beac2c8a72b080f9662897

Package: sipspa, version: v122_33_xn_asr_rls0_throttle, status: inactive
  File: bootflash:asr1000rp1-sipspa.v122_33_xn_asr_rls0_throttle.pkg, on: CC2/2
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: 6ad199569dad7d8b35beac2c8a72b080f9662897

Package: sipspa, version: v122_33_xn_asr_rls0_throttle, status: inactive
  File: bootflash:asr1000rp1-sipspa.v122_33_xn_asr_rls0_throttle.pkg, on: CC2/3
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: 6ad199569dad7d8b35beac2c8a72b080f9662897

Router#
```

In the following example, the **show version provisioned** command is entered to gather information on which sub-packages are provisioning which components on the router.

```
Router#show version provisioned
Package: Provisioning File, version: n/a, status: active
  File: bootflash:packages.conf, on: RP0
  Built: n/a, by: n/a
  File SHA1 checksum: 0b9f2c7c3d81d8455a918f285c078463c04a0cab

Package: rpbase, version: v122_33_xn_asr_rls0_throttle, status: active
  File: bootflash:asr1000rp1-rpbase.v122_33_xn_asr_rls0_throttle.pkg, on: RP0
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: 193c4810becc2a6097645f0b68f5684004bd3ab3

Package: rpaccess-k9, version: v122_33_xn_asr_rls0_throttle, status: active
  File: bootflash:asr1000rp1-rpaccess-k9.v122_33_xn_asr_rls0_throttle.pkg, on: RP0
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: 328c3d1e10f006304ce9543ab68e914b43c41b1e

Package: rpcontrol, version: v122_33_xn_asr_rls0_throttle, status: active
  File: bootflash:asr1000rp1-rpcontrol.v122_33_xn_asr_rls0_throttle.pkg, on: RP0/0
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: e4152b7fe3c2b8aca07ce1e8ad6d5a54d6d20689

Package: rpios-advipservicesk9, version: v122_33_xn_asr_rls0_throttle, status: active
  File: bootflash:asr1000rp1-rpios-advipservicesk9.v122_33_xn_asr_rls0_throttle.pkg, on:
RP0/0
  Built: 2007-11-11_17.16, by: mcpre
```

```
  File SHA1 checksum: 7f7f87f2c198c38e7b58214478c5b28ee3c7b567

Package: rpcontrol, version: v122_33_xn_asr_rls0_throttle, status: inactive
  File: bootflash:asr1000rp1-rpcontrol.v122_33_xn_asr_rls0_throttle.pkg, on: RP0/1
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: e4152b7fe3c2b8aca07ce1e8ad6d5a54d6d20689

Package: rpios-advipservicesk9, version: v122_33_xn_asr_rls0_throttle, status: inactive
  File: bootflash:asr1000rp1-rpios-advipservicesk9.v122_33_xn_asr_rls0_throttle.pkg, on:
RP0/1
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: 7f7f87f2c198c38e7b58214478c5b28ee3c7b567

Package: rpbase, version: v122_33_xn_asr_rls0_throttle, status: inactive
  File: bootflash:asr1000rp1-rpbase.v122_33_xn_asr_rls0_throttle.pkg, on: RP1
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: 193c4810becc2a6097645f0b68f5684004bd3ab3

Package: rpaccess-k9, version: v122_33_xn_asr_rls0_throttle, status: inactive
  File: bootflash:asr1000rp1-rpaccess-k9.v122_33_xn_asr_rls0_throttle.pkg, on: RP1
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: 328c3d1e10f006304ce9543ab68e914b43c41b1e

Package: rpcontrol, version: v122_33_xn_asr_rls0_throttle, status: inactive
  File: bootflash:asr1000rp1-rpcontrol.v122_33_xn_asr_rls0_throttle.pkg, on: RP1/0
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: e4152b7fe3c2b8aca07ce1e8ad6d5a54d6d20689

Package: rpios-advipservicesk9, version: v122_33_xn_asr_rls0_throttle, status: inactive
  File: bootflash:asr1000rp1-rpios-advipservicesk9.v122_33_xn_asr_rls0_throttle.pkg, on:
RP1/0
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: 7f7f87f2c198c38e7b58214478c5b28ee3c7b567

Package: rpcontrol, version: v122_33_xn_asr_rls0_throttle, status: inactive
  File: bootflash:asr1000rp1-rpcontrol.v122_33_xn_asr_rls0_throttle.pkg, on: RP1/1
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: e4152b7fe3c2b8aca07ce1e8ad6d5a54d6d20689

Package: rpios-advipservicesk9, version: v122_33_xn_asr_rls0_throttle, status: inactive
  File: bootflash:asr1000rp1-rpios-advipservicesk9.v122_33_xn_asr_rls0_throttle.pkg, on:
RP1/1
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: 7f7f87f2c198c38e7b58214478c5b28ee3c7b567

Package: rpios-advipservicesk9, version: unknown, status: active
  File: unknown, on: FP0
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: unknown

Package: rpios-advipservicesk9, version: unknown, status: inactive
  File: unknown, on: FP1
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: unknown

Package: rpios-advipservicesk9, version: unknown, status: active
  File: unknown, on: CC0
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: unknown

Package: rpios-advipservicesk9, version: unknown, status: active
  File: unknown, on: CC0/0
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: unknown
```

```
Package: rpios-advipservicesk9, version: unknown, status: active
  File: unknown, on: CC0/1
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: unknown

Package: rpios-advipservicesk9, version: unknown, status: active
  File: unknown, on: CC0/2
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: unknown

Package: rpios-advipservicesk9, version: unknown, status: inactive
  File: unknown, on: CC0/3
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: unknown

Package: rpios-advipservicesk9, version: unknown, status: active
  File: unknown, on: CC1
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: unknown

Package: rpios-advipservicesk9, version: unknown, status: active
  File: unknown, on: CC1/0
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: unknown

Package: rpios-advipservicesk9, version: unknown, status: active
  File: unknown, on: CC1/1
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: unknown

Package: rpios-advipservicesk9, version: unknown, status: active
  File: unknown, on: CC1/2
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: unknown

Package: rpios-advipservicesk9, version: unknown, status: inactive
  File: unknown, on: CC1/3
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: unknown

Package: rpios-advipservicesk9, version: unknown, status: inactive
  File: unknown, on: CC2
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: unknown

Package: rpios-advipservicesk9, version: unknown, status: inactive
  File: unknown, on: CC2/0
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: unknown

Package: rpios-advipservicesk9, version: unknown, status: inactive
  File: unknown, on: CC2/1
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: unknown

Package: rpios-advipservicesk9, version: unknown, status: inactive
  File: unknown, on: CC2/2
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: unknown

Package: rpios-advipservicesk9, version: unknown, status: inactive
  File: unknown, on: CC2/3
  Built: 2007-11-11_17.16, by: mcpre
```

```
  File SHA1 checksum: unknown

Router#
```

In the following example, the **show version running** command is entered to view which sub-packages are active on which hardware elements on the router.

```
Router#show version running
Package: Provisioning File, version: n/a, status: active
  File: bootflash:packages.conf, on: RP0
  Built: n/a, by: n/a
  File SHA1 checksum: 0b9f2c7c3d81d8455a918f285c078463c04a0cab

Package: rpbase, version: v122_33_xn_asr_rls0_throttle, status: active
  File: bootflash:asr1000rp1-rpbase.v122_33_xn_asr_rls0_throttle.pkg, on: RP0
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: 193c4810becc2a6097645f0b68f5684004bd3ab3

Package: rpaccess-k9, version: v122_33_xn_asr_rls0_throttle, status: active
  File: bootflash:asr1000rp1-rpaccess-k9.v122_33_xn_asr_rls0_throttle.pkg, on: RP0
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: 328c3d1e10f006304ce9543ab68e914b43c41b1e

Package: rpcontrol, version: v122_33_xn_asr_rls0_throttle, status: active
  File: bootflash:asr1000rp1-rpcontrol.v122_33_xn_asr_rls0_throttle.pkg, on: RP0/0
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: e4152b7fe3c2b8aca07ce1e8ad6d5a54d6d20689

Package: rpios-advipservicesk9, version: v122_33_xn_asr_rls0_throttle, status: active
  File: bootflash:asr1000rp1-rpios-advipservicesk9.v122_33_xn_asr_rls0_throttle.pkg, on:
RP0/0
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: 7f7f87f2c198c38e7b58214478c5b28ee3c7b567

Package: espbase, version: v122_33_xn_asr_rls0_throttle, status: active
  File: bootflash:asr1000rp1-espbase.v122_33_xn_asr_rls0_throttle.pkg, on: FP0
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: b1c004ed151cf60f0ce250f6ea710f43707fb010

Package: sipbase, version: v122_33_xn_asr_rls0_throttle, status: active
  File: bootflash:asr1000rp1-sipbase.v122_33_xn_asr_rls0_throttle.pkg, on: CC0
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: bd34a8a23d001f9cefcac8853a31b62ffd8272a4

Package: sipspa, version: v122_33_xn_asr_rls0_throttle, status: active
  File: bootflash:asr1000rp1-sipspa.v122_33_xn_asr_rls0_throttle.pkg, on: CC0/0
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: 6ad199569dad7d8b35beac2c8a72b080f9662897

Package: sipspa, version: v122_33_xn_asr_rls0_throttle, status: active
  File: bootflash:asr1000rp1-sipspa.v122_33_xn_asr_rls0_throttle.pkg, on: CC0/1
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: 6ad199569dad7d8b35beac2c8a72b080f9662897

Package: sipspa, version: v122_33_xn_asr_rls0_throttle, status: active
  File: bootflash:asr1000rp1-sipspa.v122_33_xn_asr_rls0_throttle.pkg, on: CC0/2
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: 6ad199569dad7d8b35beac2c8a72b080f9662897

Package: sipbase, version: v122_33_xn_asr_rls0_throttle, status: active
  File: bootflash:asr1000rp1-sipbase.v122_33_xn_asr_rls0_throttle.pkg, on: CC1
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: bd34a8a23d001f9cefcac8853a31b62ffd8272a4
```

```
Package: sipspa, version: v122_33_xn_asr_rls0_throttle, status: active
  File: bootflash:asr1000rp1-sipspa.v122_33_xn_asr_rls0_throttle.pkg, on: CC1/0
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: 6ad199569dad7d8b35beac2c8a72b080f9662897

Package: sipspa, version: v122_33_xn_asr_rls0_throttle, status: active
  File: bootflash:asr1000rp1-sipspa.v122_33_xn_asr_rls0_throttle.pkg, on: CC1/1
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: 6ad199569dad7d8b35beac2c8a72b080f9662897

Package: sipspa, version: v122_33_xn_asr_rls0_throttle, status: active
  File: bootflash:asr1000rp1-sipspa.v122_33_xn_asr_rls0_throttle.pkg, on: CC1/2
  Built: 2007-11-11_17.16, by: mcpre
  File SHA1 checksum: 6ad199569dad7d8b35beac2c8a72b080f9662897

Router#
```

Table 149 show version installed, provisioned, and running Field Descriptions

Field	Description
`Package:`	The individual sub-package name.
`version:`	The consolidated package version of the individual sub-package.
`status:`	Reveals if the sub-package is active or inactive for the specific hardware component only.
`File:`	The location and filename of the individual sub-package file.
`on:`	The hardware component.
`Built:`	The date the individual sub-package was built.
`File SHA1 checksum:`	The SHA1 sum for the file. This sum can be compared against a SHA1 sum generated by any SHA1 sum-generating tool.

Related Commands

Command	Description
show diag	Displays hardware and diagnostic information for a networking device, a line card, a processor, a jacket card, a chassis, or a network module.
show inventory	Displays the Cisco Unique Device Identifier information, including the Product ID, the Version ID, and the Serial Number, for the hardware device and hardware components.

show warm-reboot

To display the statistics for attempted warm reboots, use the **show warm-reboot** command in privileged EXEC mode.

show warm-reboot

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.3(2)T	This command was introduced.
12.2(18)S	This command was integrated into Cisco IOS Relase 12.2(18)S.
12.2(28)SB	This command was integrated into Cisco IOS Relase 12.2(28)SB.

Usage Guidelines

Use the **show warm-reboot** command to see if warm rebooting is enabled, and, if so, how many warm reloads have occurred and how much space in kilobytes (KB) is consumed by warm-reboot storage, which is the RAM area used to store the data segment that enables warm reloading to function.

Examples

The following example is sample output from the **show warm-reboot** command:

```
Router# show warm-reboot

Warm Reboot is enabled

Statistics:
10 warm reboots have taken place since the last cold reboot
XXX KB taken up by warm reboot storage
```

Related Commands

Command	Description
warm-reboot	Enables a router to warm-reboot.

show whoami

To display information about the terminal line of the current user, including host name, line number, line speed, and location, use the **show whoami** command in EXEC mode.

show whoami [*text*]

Syntax Description

text	(Optional) Additional data to print to the screen.

Command Modes

EXEC

Command History

Release	Modification
10.0	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

If text is included as an argument in the command, that text is displayed as part of the additional data about the line.

To prevent the information from being lost if the menu display clears the screen, this command always displays a --More-- prompt before returning. Press the space bar to return to the prompt.

Examples

The following example is sample output from the **show whoami** command:

```
Router> show whoami


Comm Server "Router", Line 0 at 0bps.  Location "Second floor, West"

--More--
Router>
```

showmon

To show both the ReadOnly and the Upgrade ROMmon image versions when you are in ROMmon mode, as well as which ROMmon image is running on the Cisco 7200 VXR or Cisco 7301 router, use the **showmon** command in ROM monitor mode.

showmon

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values

Command Modes

ROM monitor mode

Command History

Release	Modification
12.0(28)S	This command was introduced on the Cisco 7200 VXR router. It was introduced in ROMmon version 12.3(4r)T1 for the Cisco 7200 VXR router.
12.3(8)T	This command was integrated into Cisco IOS Release 12.3(8)T and supported on the Cisco 7200 VXR router and Cisco 7301 router. It was introduced in ROMmon version 12.3(4r)T2 for the Cisco 7301 router.
12.3(9)	This command was integrated into Cisco IOS Release 12.3(9) and supported on the Cisco 7200 VXR router and Cisco 7301 router.

Usage Guidelines

Use the **showmon** command when you are in ROM monitor mode. Use the **show rom-monitor** command when you are in Cisco IOS.

Examples

The following example, applicable to both the Cisco 7200 VXR and Cisco 7301 routers, uses the **showmon** command in ROMmon to display both ROMmon images and to verify that the Upgrade ROMmon image is running:

```
rommon 1 > showmon

ReadOnly ROMMON version is:
System Bootstrap, Version 12.2(20031011:151758) [biff]
Copyright (c) 2004 by Cisco Systems, Inc.


Upgrade ROMMON version is:
System Bootstrap, Version 12.2(20031011:151758) [biff]
Copyright (c) 2004 by Cisco Systems, Inc.

Upgrade ROMMON currently running
Upgrade ROMMON is selected for next boot
rommon 2 >
```

Related Commands

Command	Description
rommon-pref	Selects a ReadOnly or Upgrade ROMmon image to be booted on the next reload of a Cisco 7200 VXR or Cisco 7301 when you are in ROMmon.